Amazon Elastic Compute Cloud

User Guide for Microsoft Windows

API Version 2015-04-15

Amazon Elastic Compute Cloud: User Guide for Microsoft Windows

Table of Contents

What Is Amazon EC2? ... 1
- Features of Amazon EC2 ... 1
- How to Get Started with Amazon EC2 ... 2
- Related Services ... 3
- Accessing Amazon EC2 ... 3
- Pricing for Amazon EC2 ... 4
- Basic Infrastructure ... 5
  - Amazon Machine Images and Instances ... 5
  - Regions and Availability Zones ... 6
  - Storage ... 6
  - Root Device Volume ... 8
  - Networking and Security ... 10
  - AWS Identity and Access Management ... 10
  - Differences between Windows Server and an Amazon EC2 Windows Instance ... 11
  - Designing Your Applications to Run on Amazon EC2 Windows Instances ... 12

Setting Up ... 14
- Sign Up for AWS ... 14
- Create an IAM User ... 15
- Create a Key Pair ... 16
- Create a Virtual Private Cloud (VPC) ... 17
- Create a Security Group ... 17

Getting Started: Launch and Connect ... 20
- Overview ... 20
- Launch a Windows Instance ... 21
- Connect to Your Windows Instance ... 23
- Create a CloudWatch Alarm to Monitor Your Instance ... 24
- Clean Up ... 26

Best Practices ... 28

Tutorials ... 30
- Tutorial: Deploy a WordPress Blog ... 30
  - Prerequisites ... 30
  - Installing the Microsoft Web Platform Installer ... 31
  - Installing WordPress ... 31
  - Configure Security Keys ... 32
  - Administrative Information ... 33
  - Making Your WordPress Site Public ... 34
- Tutorial: Installing a WAMP Server ... 34
- Tutorial: Installing a WIMP Server ... 37
- Tutorial: Set Up a Windows HPC Cluster ... 41
  - Prerequisites ... 41
  - Task 1: Set Up Your Active Directory Domain Controller ... 41
  - Task 2: Configure Your Head Node ... 43
  - Task 3: Set Up the Compute Node ... 45
  - Task 4: Scale Your HPC Compute Nodes (Optional) ... 46
  - Running the Lizard Performance Measurement Application ... 47
  - Create_AD_security.bat ... 47
  - Create-HPC-sec-group.bat ... 48

Amazon Machine Images ... 50
- Using an AMI ... 50
- Creating Your Own AMI ... 51
- Buying, Sharing, and Selling AMIs ... 51
- Deregistering Your AMI ... 51
- AWS Windows AMIs ... 51
  - Selecting an Initial Windows AMI ... 51
  - Keeping Your AMIs Up-to-Date ... 52

AMI Types ... 52
Launch Permissions ... 53
Storage for the Root Device ... 53
Finding a Windows AMI ... 56
Finding a Windows AMI Using the Amazon EC2 Console ... 56
Finding an AMI Using the AWS CLI ... 57
Finding an AMI Using the Amazon EC2 CLI ... 57
Finding an AMI Using the AWS Tools for Windows PowerShell ... 57
Finding a Windows Server 2003 AMI ... 58
Shared AMIs ... 58
Finding Shared AMIs ... 58
Making an AMI Public ... 60
Sharing an AMI with Specific AWS Accounts ... 62
Using Bookmarks ... 63
Guidelines for Shared Windows AMIs ... 64
Paid AMIs ... 64
Selling Your AMI ... 65
Finding a Paid AMI ... 65
Purchase a Paid AMI ... 66
Getting the Product Code for Your Instance ... 67
Using Paid Support ... 67
Bills for Paid and Supported AMIs ... 67
Managing Your AWS Marketplace Subscriptions ... 68
Creating an Amazon EBS-Backed Windows AMI ... 68
Creating an AMI from an Instance ... 68
Creating an Instance Store-Backed Windows AMI ... 70
Instance Store-Backed Windows AMIs ... 70
Preparing to Create an Instance Store-Backed Windows AMI ... 71
Bundling an Instance Store-Backed Windows Instance ... 72
Registering an Instance Store-Backed Windows AMI ... 73
Copying an AMI ... 74
AMI Copy ... 74
Copying an Amazon EC2 AMI ... 75
Stopping a Pending AMI Copy Operation ... 76
Deregistering Your AMI ... 76
Cleaning Up Your Amazon EBS-Backed AMI ... 76
Cleaning Up Your Instance Store-Backed AMI ... 77
Windows AMI Versions ... 78
Configuration Settings and Drivers ... 78
Updating Your Windows Instance ... 79
Determining Your Instance Version ... 79
AWS Windows AMI Versions ... 80
Image Changes ... 88
Subscribing to Windows AMI Notifications ... 89
Upgrading or Migrating a Windows Server Instance ... 90
Create a Standard Amazon Machine Image Using Sysprep ... 90
Before You Begin ... 90
Using Sysprep with the EC2Config Service ... 90
Run Sysprep with the EC2Config Service ... 94
Troubleshooting Sysprep with EC2Config ... 94
Instances ... 96
Instance Types ... 96
Available Instance Types ... 97
Hardware Specifications ... 98
Networking and Storage Features ... 98
Instance Limits ... 99
T2 Instances ... 99
C4 Instances ... 102

GPU Instances ... 104
I2 Instances ... 106
D2 Instances ... 107
HI1 Instances ... 108
HS1 Instances ... 110
T1 Micro Instances ... 111
Resizing Instances ... 118
Spot Instances ... 122
Concepts ... 122
How to Get Started ... 122
Related Services ... 123
Pricing ... 124
How Spot Instances Work ... 124
How Spot Fleet Works ... 127
Spot Instance Pricing History ... 132
Spot Instance Requests ... 133
Spot Fleet Requests ... 138
Spot Bid Status ... 150
Spot Instance Interruptions ... 155
Spot Instance Data Feed ... 156
Spot Instance Limits ... 159
Instance Metadata and User Data ... 160
Retrieving Instance Metadata ... 161
Adding User Data ... 163
Retrieving User Data ... 163
Retrieving Dynamic Data ... 164
Instance Metadata Categories ... 164
Importing and Exporting Instances ... 169
Prerequisites ... 169
Importing a VM into Amazon EC2 Using ImportImage ... 177
Importing a VM into Amazon EC2 Using ImportInstance ... 185
Exporting Amazon EC2 Instances ... 195
Troubleshooting ... 197
Instance Lifecycle ... 203
Instance Launch ... 203
Instance Stop and Start (Amazon EBS-backed instances only) ... 204
Instance Reboot ... 204
Instance Retirement ... 204
Instance Termination ... 205
Differences Between Reboot, Stop, and Terminate ... 205
Launch ... 206
Launching an Instance ... 207
Launching an Instance From an Existing Instance ... 212
Launching an AWS Marketplace Instance ... 213
Connect ... 216
Prerequisites ... 216
Connecting to Windows ... 216
Transfer Files to Windows Server Instances ... 218
Stop and Start ... 218
Overview ... 219
Stopping and Starting Your Instances ... 220
Modifying a Stopped Instance ... 221
Troubleshooting ... 221
Reboot ... 222
Retire ... 222
Identifying Instances Scheduled for Retirement ... 222
Working with Instances Scheduled for Retirement ... 223
Terminate ... 224

Instance Termination ... 225
Terminating an Instance ... 225
Enabling Termination Protection ... 226
Changing the Shutdown Behavior ... 227
Preserving Amazon EBS Volumes on Instance Termination ... 227
Recover ... 229
OS Upgrade ... 231
Configure Instances ... 235
Using EC2Config ... 235
Overview of EC2Config Tasks ... 236
Ec2 Service Properties ... 237
EC2Config Settings Files ... 240
Executing User Data ... 243
Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs ... 245
Installing the Latest Version of EC2Config ... 256
Stopping, Restarting, Deleting, or Uninstalling EC2Config ... 258
PV Drivers ... 258
Drivers According to Windows Version ... 262
AWS PV Drivers ... 262
Citrix PV Drivers ... 264
RedHat PV Drivers ... 264
Related Topics ... 264
Upgrading PV Drivers ... 265
Troubleshooting PV Drivers ... 270
Setting the Password ... 277
Changing the Administrator Password After Connecting ... 277
Resetting an Administrator Password that's Lost or Expired ... 278
Setting the Time ... 281
Changing the Time Zone ... 281
Configuring Network Time Protocol (NTP) ... 282
Configuring Time Settings for Windows Server 2008 and later ... 283
Configuring Time Settings for Windows Server 2003 ... 284
Related Topics ... 284
Managing Configuration ... 284
Grant IAM Users Access to SSM ... 285
Prepare the Instance ... 285
Create the JSON File ... 286
Create the Configuration Document ... 288
Associate the Configuration Document with the Instance ... 288
Manually Apply the Configuration ... 289
Disassociate the Configuration Document from the Instance ... 289
Delete the Configuration Document ... 289
Troubleshooting ... 290
Joining an AWS Domain ... 291
Limitations ... 292
Prerequisites ... 292
Joining a Domain Using the AWS CLI or AWS Tools for Windows PowerShell ... 293
Joining a Domain Using the Amazon EC2 Launch Wizard ... 295
Getting the Domain Join Status ... 296
Connecting To Your Instance Using Domain Credentials ... 297
Troubleshooting ... 297
Viewing Information About Your Associations ... 298
Changing an Association ... 299
Deleting a Configuration Document ... 299
Sending Log Data to CloudWatch ... 299
Step 1: Prepare Your Instance ... 300
Step 2: Create a JSON File ... 301
Step 3: Configure the Region and Namespace for CloudWatch and CloudWatch Logs ... 304

Step 4: Configure the Performance Counters and Logs to Send to CloudWatch and CloudWatch Logs ... 305
Step 5: Configure the Flow Control ... 311
Step 6: Create a Configuration Document ... 311
Step 7: Associate the Configuration Document with the Instance ... 312
Configuring a Secondary Private IP Address ... 312
Prerequisites ... 313
Step 1: Configure Static IP Addressing on Your Windows Instance ... 313
Step 2: Configure a Secondary Private IP Address for Your Windows Instance ... 315
Step 3: Configure Applications to Use the Secondary Private IP Address ... 316
Monitoring ... 318
Automated and Manual Monitoring ... 320
Automated Monitoring Tools ... 320
Manual Monitoring Tools ... 321
Best Practices for Monitoring ... 321
Monitoring the Status of Your Instances ... 322
Instance Status Checks ... 322
Scheduled Events ... 326
Monitoring Your Instances with CloudWatch ... 330
Enabling or Disabling Detailed Monitoring on an Amazon EC2 Instance ... 331
View Amazon EC2 Metrics ... 334
Get Statistics for Metrics ... 340
Graphing Metrics ... 357
Create a CloudWatch Alarm ... 361
Create Alarms That Stop, Terminate, Reboot, or Recover an Instance ... 368
Network and Security ... 393
Key Pairs ... 394
Creating Your Key Pair Using Amazon EC2 ... 394
Importing Your Own Key Pair to Amazon EC2 ... 395
Retrieving the Public Key for Your Key Pair on Windows ... 397
Verifying Your Key Pair's Fingerprint ... 397
Deleting Your Key Pair ... 398
Security Groups ... 398
Security Groups for EC2-Classic ... 399
Security Groups for EC2-VPC ... 399
Security Group Rules ... 400
Default Security Groups ... 401
Custom Security Groups ... 401
Creating a Security Group ... 402
Describing Your Security Groups ... 403
Adding Rules to a Security Group ... 403
Deleting Rules from a Security Group ... 404
Deleting a Security Group ... 405
API and Command Overview ... 405
Controlling Access ... 406
Network Access to Your Instance ... 406
Amazon EC2 Permission Attributes ... 406
IAM and Amazon EC2 ... 407
IAM Policies ... 408
IAM Roles ... 442
Network Access ... 448
Amazon VPC ... 449
Benefits of Using a VPC ... 450
Differences Between EC2-Classic and EC2-VPC ... 450
Sharing and Accessing Resources Between EC2-Classic and EC2-VPC ... 453
Instance Types Available Only in a VPC ... 455
Amazon VPC Documentation ... 455
Supported Platforms ... 455

ClassicLink ... 457
Migrating from EC2-Classic to a VPC ... 465
Instance IP Addressing ... 474
Private IP Addresses and Internal DNS Hostnames ... 475
Public IP Addresses and External DNS Hostnames ... 475
Elastic IP Addresses ... 476
Amazon DNS Server ... 477
IP Address Differences Between EC2-Classic and EC2-VPC ... 477
Determining Your Public, Private, and Elastic IP Addresses ... 477
Assigning a Public IP Address ... 479
Multiple Private IP Addresses ... 480
Elastic IP Addresses ... 485
Elastic IP Addresses in EC2-Classic ... 485
Elastic IP Addresses in a VPC ... 485
Elastic IP Address Differences Between EC2-Classic and EC2-VPC ... 486
Migrating an Elastic IP Address from EC2-Classic to EC2-VPC ... 487
Working with Elastic IP Addresses ... 487
Using Reverse DNS for Email Applications ... 491
Elastic IP Address Limit ... 491
Elastic Network Interfaces ... 492
Private IP Addresses Per ENI Per Instance Type ... 493
Creating a Management Network ... 495
Use Network and Security Appliances in Your VPC ... 495
Creating Dual-homed Instances with Workloads/Roles on Distinct Subnets ... 495
Create a Low Budget High Availability Solution ... 496
Monitoring IP Traffic on Your Network Interface ... 496
Best Practices for Configuring Elastic Network Interfaces ... 496
Creating an Elastic Network Interface ... 496
Deleting an Elastic Network Interface ... 497
Viewing Details about an Elastic Network Interface ... 497
Attaching an Elastic Network Interface When Launching an Instance ... 498
Attaching an Elastic Network Interface to a Stopped or Running Instance ... 499
Detaching an Elastic Network Interface from an Instance ... 500
Changing the Security Group of an Elastic Network Interface ... 500
Changing the Source/Destination Checking of an Elastic Network Interface ... 501
Associating an Elastic IP Address with an Elastic Network Interface ... 502
Disassociating an Elastic IP Address from an Elastic Network Interface ... 502
Changing Termination Behavior for an Elastic Network Interface ... 503
Adding or Editing a Description for an Elastic Network Interface ... 503
Adding or Editing Tags for an Elastic Network Interface ... 504
Placement Groups ... 504
Placement Group Limitations ... 505
Launching Instances into a Placement Group ... 505
Deleting a Placement Group ... 507
Network MTU ... 507
Jumbo Frames (9001 MTU) ... 508
Path MTU Discovery ... 508
Check the Path MTU Between Two Hosts ... 509
Check and Set the MTU on your Amazon EC2 Instance ... 509
Troubleshooting ... 510
Enabling Enhanced Networking ... 510
Instances that Support Enhanced Networking ... 511
Requirements ... 511
Testing Whether Enhanced Networking Is Enabled ... 511
Enabling Enhanced Networking on Windows ... 513
Storage ... 515
Amazon EBS ... 516
Features of Amazon EBS ... 517

EBS Volumes ... 518
EBS Snapshots ... 549
EBS Optimization ... 555
EBS Encryption ... 558
EBS Performance ... 561
EBS Commands ... 574
Instance Store ... 577
Instance Store Lifetime ... 577
Instance Store Volumes ... 578
Add Instance Store Volumes ... 579
SSD Instance Store Volumes ... 582
Amazon S3 ... 583
Amazon S3 and Amazon EC2 ... 583
Instance Volume Limits ... 585
Linux-Specific Volume Limits ... 585
Windows-Specific Volume Limits ... 585
Bandwidth vs Capacity ... 585
Device Naming ... 586
Available Device Names ... 586
Device Name Considerations ... 586
Block Device Mapping ... 587
Block Device Mapping Concepts ... 587
AMI Block Device Mapping ... 590
Instance Block Device Mapping ... 592
Mapping Disks to Volumes ... 596
Listing the Disks Using Windows Disk Management ... 597
Listing the Disks Using Windows PowerShell ... 598
Disk Device to Device Name Mapping ... 600
Using Public Data Sets ... 602
Public Data Set Concepts ... 602
Finding Public Data Sets ... 602
Creating a Public Data Set Volume from a Snapshot ... 603
Attaching and Mounting the Public Data Set Volume ... 604
Resources and Tags ... 605
Resource Locations ... 605
Listing and Filtering Your Resources ... 606
Advanced Search ... 606
Listing Resources Using the Console ... 607
Filtering Resources Using the Console ... 608
Listing and Filtering Using the CLI and API ... 609
Tagging Your Resources ... 609
Tag Basics ... 610
Tag Restrictions ... 610
Tagging Your Resources for Billing ... 612
Working with Tags Using the Console ... 612
Working with Tags Using the CLI or API ... 617
Service Limits ... 618
Viewing Your Current Limits ... 618
Requesting a Limit Increase ... 619
Usage Reports ... 619
Available Reports ... 619
Getting Set Up for Usage Reports ... 619
Granting IAM Users Access to the Amazon EC2 Usage Reports ... 621
Instance Usage ... 621
Reserved Instance Utilization ... 625
AWS Systems Manager for Microsoft System Center VMM ... 630
Features ... 630
Limitations ... 631

Requirements 631
Getting Started 631
Setting Up 631
Sign Up for AWS 631
Set Up Access for Users 632
Deploy the Add-In 634
Provide Your AWS Credentials 634
Managing EC2 Instances 635
Creating an EC2 Instance 635
Viewing Your Instances 637
Connecting to Your Instance 638
Rebooting Your Instance 638
Stopping Your Instance 639
Starting Your Instance 639
Terminating Your Instance 639
Importing Your VM 639
Prerequisites 640
Importing Your Virtual Machine 640
Checking the Import Task Status 641
Backing Up Your Imported Instance 641
Troubleshooting 642
Error: Add-in cannot be installed 642
Installation Errors 642
Checking the Log File 643
Errors Importing a VM 643
Uninstalling the Add-In 643
AWS Management Pack 644
Overview of AWS Management Pack for System Center 2012 645
Overview of AWS Management Pack for System Center 2007 R2 646
Downloading 647
System Center 2012 647
System Center 2007 R2 648
Deploying 648
Step 1: Installing the AWS Management Pack 649
Step 2: Configuring the Watcher Node 650
Step 3: Create an AWS Run As Account 651
Step 4: Run the Add Monitoring Wizard 653
Step 5: Configure Ports and Endpoints 657
Using 658
Views 658
Discoveries 672
Monitors 673
Rules 673
Events 674
Health Model 675
Customizing the AWS Management Pack 676
Upgrading 677
System Center 2012 677
System Center 2007 R2 677
Uninstalling 678
System Center 2012 678
System Center 2007 R2 678
Troubleshooting 678
Errors 4101 and 4105 679
Error 4513 679
Event 623 679
Events 2023 and 2120 680
Event 6024 680

General Troubleshooting for System Center 2012 — Operations Manager 680
General Troubleshooting for System Center 2007 R2 681
AWS Diagnostics for Microsoft Windows Server 682
Analysis Rules 682
Analyzing the Current Instance 683
Collecting Data From an Offline Instance 685
Data File Storage 685
Troubleshooting 687
Boot an EC2 Windows Instance into Directory Services Restore Mode (DSRM) 688
Driver Support for DSRM 688
Configure an Instance to Boot into DSRM 689
High CPU shortly after Windows starts 690
No console output 690
Instance terminates immediately 691
"Password is not available" 691
"Password not available yet" 692
"Cannot retrieve Windows password" 692
"Waiting for the metadata service" 693
Remote Desktop can't connect to the remote computer 695
RDP displays a black screen instead of the desktop 697
"Unable to activate Windows" 698
"Windows is not genuine (0x80070005)" 699
"No Terminal Server License Servers available to provide a license" 699
Instance loses network connectivity or scheduled tasks don't run when expected 699
Insufficient Instance Capacity 700
Instance Limit Exceeded 700
Windows Server 2012 R2 not available on the network 700
Document History 701
AWS Glossary 714

What Is Amazon EC2?

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

For more information about cloud computing, see What is Cloud Computing?

Features of Amazon EC2

Amazon EC2 provides the following features:

- Virtual computing environments, known as *instances*
- Preconfigured templates for your instances, known as *Amazon Machine Images (AMIs)*, that package the bits you need for your server (including the operating system and additional software)
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as *instance types*
- Secure login information for your instances using *key pairs* (AWS stores the public key, and you store the private key in a secure place)
- Storage volumes for temporary data that's deleted when you stop or terminate your instance, known as *instance store volumes*
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as *Amazon EBS volumes*
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as *regions* and *Availability Zones*
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using *security groups*
- Static IP addresses for dynamic cloud computing, known as *Elastic IP addresses*
- Metadata, known as *tags*, that you can create and assign to your Amazon EC2 resources
- Virtual networks you can create that are logically isolated from the rest of the AWS cloud, and that you can optionally connect to your own network, known as *virtual private clouds* (VPCs)

For more information about the features of Amazon EC2, see the Amazon EC2 product page.

Amazon EC2 enables you to run any compatible Windows-based solution on our high-performance, reliable, cost-effective, cloud computing platform. For more information, see Amazon EC2 Running Windows Server & SQL.

For more information about running your website on AWS, see Websites & Website Hosting.

How to Get Started with Amazon EC2

The first thing you need to do is get set up to use Amazon EC2. After you are set up, you are ready to complete the Getting Started tutorial for Amazon EC2. Whenever you need more information about a feature of Amazon EC2, you can read the technical documentation.

Get Up and Running

- Setting Up with Amazon EC2 (p. 14)
- Getting Started with Amazon EC2 Windows Instances (p. 20)

Basics

- Amazon EC2 Basic Infrastructure for Windows (p. 5)
- Instance Types (p. 96)
- Tags (p. 609)

Networking and Security

- Amazon EC2 Key Pairs and Windows Instances (p. 394)
- Security Groups (p. 398)
- Elastic IP Addresses (p. 485)
- Amazon EC2 and Amazon VPC (p. 449)

Storage

- Amazon EBS (p. 516)
- Instance Store (p. 577)

Working with Windows Instances

- Differences between Windows Server and an Amazon EC2 Windows Instance (p. 11)
- Designing Your Applications to Run on Amazon EC2 Windows Instances (p. 12)
- Getting Started with AWS: Hosting a .NET Web App

If you have questions about whether AWS is right for you, contact AWS Sales. If you have technical questions about Amazon EC2, use the Amazon EC2 forum.

Related Services

You can provision Amazon EC2 resources, such as instances and volumes, directly using Amazon EC2. You can also provision Amazon EC2 resources using other services in AWS. For more information, see the following documentation:

- Auto Scaling Developer Guide
- AWS CloudFormation User Guide
- AWS Elastic Beanstalk Developer Guide
- AWS OpsWorks User Guide

To automatically distribute incoming application traffic across multiple instances, use Elastic Load Balancing. For more information, see Elastic Load Balancing Developer Guide.

To monitor basic statistics for your instances and Amazon EBS volumes, use Amazon CloudWatch. For more information, see the Amazon CloudWatch Developer Guide.

To monitor the calls made to the Amazon EC2 API for your account, including calls made by the AWS Management Console, command line tools, and other services, use AWS CloudTrail. For more information, see the AWS CloudTrail User Guide.

To get a managed relational database in the cloud, use Amazon Relational Database Service (Amazon RDS) to launch a database instance. Although you can set up a database on an EC2 instance, Amazon RDS offers the advantage of handling your database management tasks, such as patching the software, backing up, and storing the backups. For more information, see Amazon Relational Database Service Developer Guide.

Accessing Amazon EC2

Amazon EC2 provides a web-based user interface, the Amazon EC2 console. If you've signed up for an AWS account, you can access the Amazon EC2 console by signing into the AWS Management Console and selecting **EC2** from the console home page.

If you prefer to use a command line interface, you have several options:

AWS Command Line Interface (CLI)
: Provides commands for a broad set of AWS products, and is supported on Windows, Mac, and Linux. To get started, see AWS Command Line Interface User Guide. For more information about the commands for Amazon EC2, see ec2 in the *AWS Command Line Interface Reference*.

Amazon EC2 Command Line Interface (CLI) Tools
: Provides commands for Amazon EC2, Amazon EBS, and Amazon VPC, and is supported on Windows, Mac, and Linux. To get started, see Setting Up the Amazon EC2 Command Line Interface Tools on Windows and Commands (CLI Tools) in the *Amazon EC2 Command Line Reference*.

AWS Tools for Windows PowerShell
: Provides commands for a broad set of AWS products for those who script in the PowerShell environment. To get started, see the AWS Tools for Windows PowerShell User Guide. For more information about the cmdlets for Amazon EC2, see the AWS Tools for Windows PowerShell Reference.

Amazon EC2 provides a Query API. These requests are HTTP or HTTPS requests that use the HTTP verbs GET or POST and a Query parameter named `Action`. For more information about the API actions for Amazon EC2, see Actions in the *Amazon EC2 API Reference*.

If you prefer to build applications using language-specific APIs instead of submitting a request over HTTP or HTTPS, AWS provides libraries, sample code, tutorials, and other resources for software developers. These libraries provide basic functions that automate tasks such as cryptographically signing your requests, retrying requests, and handling error responses, making it is easier for you to get started. For more information, see AWS SDKs and Tools.

Pricing for Amazon EC2

When you sign up for AWS, you can get started with Amazon EC2 for free using the AWS Free Tier.

Amazon EC2 provides the following purchasing options for instances:

On-Demand instances
: Pay for the instances that you use by the hour, with no long-term commitments or up-front payments.

Reserved Instances
: Make a low, one-time, up-front payment for an instance, reserve it for a one- or three-year term, and pay a significantly lower hourly rate for these instances.

Spot instances
: Specify the maximum hourly price that you are willing to pay to run a particular instance type. The Spot price fluctuates based on supply and demand, but you never pay more than the maximum price you specified. If the Spot price moves higher than your maximum price, Amazon EC2 shuts down your Spot instances.

For a complete list of charges and specific prices for Amazon EC2, see Amazon EC2 Pricing.

To calculate the cost of a sample provisioned environment, see AWS Economics Center.

To see your bill, go to your AWS Account Activity page. Your bill contains links to usage reports that provide details about your bill. To learn more about AWS account billing, see AWS Account Billing.

If you have questions concerning AWS billing, accounts, and events, contact AWS Support.

For an overview of Trusted Advisor, a service that helps you optimize the costs, security, and performance of your AWS environment, see AWS Trusted Advisor.

Amazon EC2 Basic Infrastructure for Windows

As you get started with Amazon EC2, you'll benefit from understanding the components of its basic infrastructure and how they compare or contrast with your own data centers.

Concepts

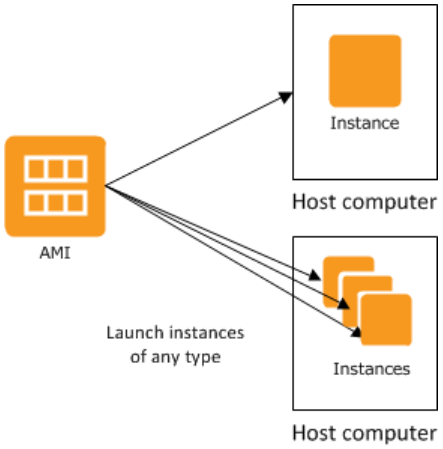
- Amazon Machine Images and Instances (p. 5)
- Regions and Availability Zones (p. 6)
- Storage (p. 6)
- Root Device Volume (p. 8)
- Networking and Security (p. 10)
- AWS Identity and Access Management (p. 10)
- Differences between Windows Server and an Amazon EC2 Windows Instance (p. 11)
- Designing Your Applications to Run on Amazon EC2 Windows Instances (p. 12)

Amazon Machine Images and Instances

An *Amazon Machine Image (AMI)* is a template that contains a software configuration (for example, an operating system, an application server, and applications). From an AMI, you launch *instances*, which are copies of the AMI running as virtual servers in the cloud.

Amazon publishes many AMIs that contain common software configurations for public use. In addition, members of the AWS developer community have published their own custom AMIs. You can also create your own custom AMI or AMIs; doing so enables you to quickly and easily start new instances that have everything you need. For example, if your application is a website or web service, your AMI could include a web server, the associated static content, and the code for the dynamic pages. As a result, after you launch an instance from this AMI, your web server starts, and your application is ready to accept requests.

You can launch different types of instances from a single AMI. An *instance type* essentially determines the hardware of the host computer used for your instance. Each instance type offers different compute and memory facilities. Select an instance type based on the amount of memory and computing power that you need for the applications or software that you plan to run on the instance. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances. You can also launch multiple instances from an AMI, as shown in the following figure.



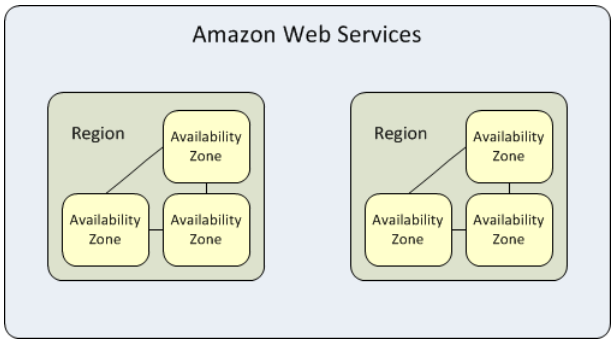
Your Windows instances keep running until you stop or terminate them, or until they fail. If an instance fails, you can launch a new one from the AMI.

Your AWS account has a limit on the number of instances that you can have running. For more information about this limit, and how to request an increase, see How many instances can I run in Amazon EC2 in the Amazon EC2 General FAQ.

Regions and Availability Zones

Amazon has data centers in different areas of the world (for example, North America, Europe, and Asia). Correspondingly, Amazon EC2 is available to use in different *regions*. By launching instances in separate regions, you can design your application to be closer to specific customers or to meet legal or other requirements. Prices for Amazon EC2 usage vary by region (for more information about pricing by region, see Amazon EC2 Pricing).

Each region contains multiple distinct locations called *Availability Zones*. Each Availability Zone is engineered to be isolated from failures in other Availability Zones, and to provide inexpensive, low-latency network connectivity to other zones in the same region. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location.



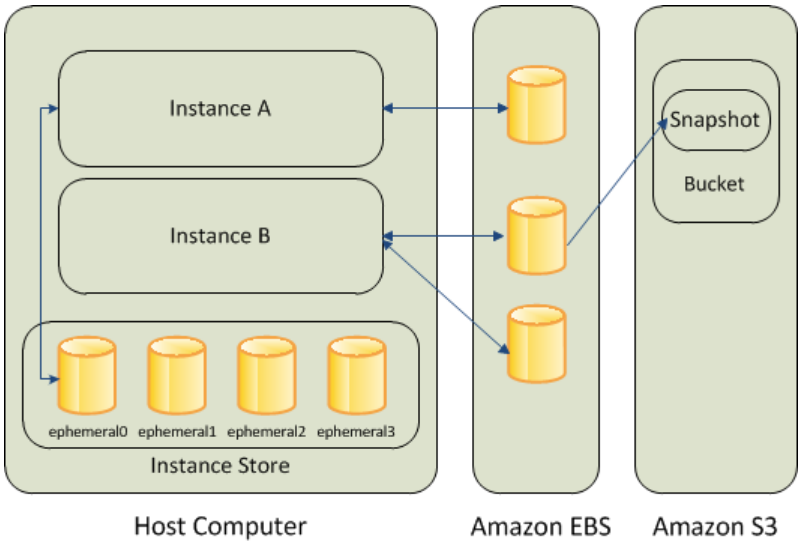
For more information about the available regions and Availability Zones, see Using Regions and Availability Zones in the *Amazon EC2 User Guide for Linux Instances*.

Storage

When using Amazon EC2, you may have data that you need to store. Amazon EC2 offers the following storage options:

- Amazon Elastic Block Store (Amazon EBS)
- Amazon EC2 Instance Store (p. 577)
- Amazon Simple Storage Service (Amazon S3)

The following figure shows the relationship between these types of storage.

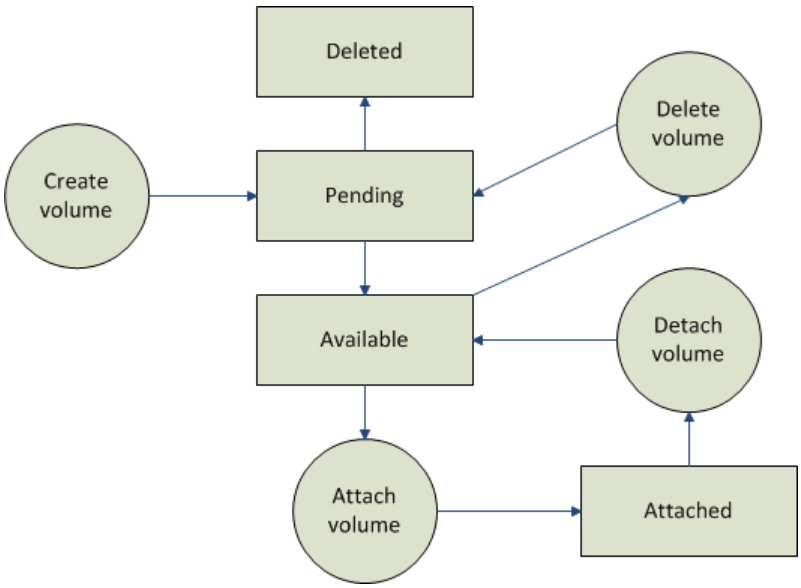


Amazon EBS Volumes

Amazon EBS volumes are the recommended storage option for the majority of use cases. Amazon EBS provides your instances with persistent, block-level storage. Amazon EBS volumes are essentially hard disks that you can attach to a running instance.

Amazon EBS is especially suited for applications that require a database, a file system, or access to raw block-level storage.

As illustrated in the previous figure, you can attach multiple volumes to an instance. Also, to keep a backup copy of your data, you can create a *snapshot* of an EBS volume, which is stored in Amazon S3. You can create a new Amazon EBS volume from a snapshot, and attach it to another instance. You can also detach a volume from an instance and attach it to a different instance. The following figure illustrates the life cycle of an EBS volume.



For more information about Amazon EBS volumes, see Amazon Elastic Block Store (Amazon EBS) (p. 516).

Instance Store

All instance types, with the exception of Micro instances, offer *instance store*, which provides your instances with temporary, block-level storage. This is storage that is physically attached to the host computer. The data on an instance store volume doesn't persist when the associated instance is stopped or terminated. For more information about instance store volumes, see Amazon EC2 Instance Store (p. 577).

Instance store is an option for inexpensive temporary storage. You can use instance store volumes if you don't require data persistence.

Amazon S3

Amazon S3 is storage for the Internet. It provides a simple web service interface that enables you to store and retrieve any amount of data from anywhere on the web. For more information about Amazon S3, see the Amazon S3 product page.

Root Device Volume

When you launch an instance, the *root device volume* contains the image used to boot the instance. You can launch an Amazon EC2 Windows instance using an AMI backed either by instance store or by Amazon Elastic Block Store (Amazon EBS).

- **Instances launched from an AMI backed by Amazon EBS** use an Amazon EBS volume as the root device. The root device volume of an Amazon EBS-backed AMI is an Amazon EBS snapshot. When an instance is launched using an Amazon EBS-backed AMI, a root EBS volume is created from the EBS snapshot and attached to the instance. The root device volume is then used to boot the instance.
- **Instances launched from an AMI backed by instance store** use an instance store volume as the root device. The image of the root device volume of an instance store-backed AMI is initially stored in Amazon S3. When an instance is launched using an instance store-backed AMI, the image of its root device is copied from Amazon S3 to the root partition of the instance. The root device volume is then used to boot the instance.

  Important
  The only Windows AMIs that can be backed by instance store are those for Windows Server 2003. Instance store-backed instances don't have the available disk space required for later versions of Windows Server.

For a summary of the differences between instance store-backed AMIs and Amazon EBS-backed AMIs, see Storage for the Root Device (p. 53).

Determining the Root Device Type of an AMI

You can determine the root device type of an AMI using the console or the command line.

To determine the root device type of an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**, and select the AMI.
3. Check the value of **Root Device Type** in the **Details** tab as follows:
   - If the value is `ebs`, this is an Amazon EBS-backed AMI.
   - If the value is `instance store`, this is an instance store-backed AMI.

To determine the root device type of an AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-images (AWS CLI)
- ec2-describe-images (Amazon EC2 CLI)
- Get-EC2Image (AWS Tools for Windows PowerShell)

Determining the Root Device Type of an Instance

You can determine the root device type of an instance using the console or the command line.

To determine the root device type of an instance using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**, and select the instance.
3. Check the value of **Root device type** in the **Description** tab as follows:
   - If the value is `ebs`, this is an Amazon EBS-backed instance.
   - If the value is `instance store`, this is an instance store-backed instance.

To determine the root device type of an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-instances (AWS CLI)
- ec2-describe-instances (Amazon EC2 CLI)
- Get-EC2Instance (AWS Tools for Windows PowerShell)

Changing the Root Device Volume to Persist

Using the console, you can change the `DeleteOnTermination` attribute when you launch an instance. To change this attribute for a running instance, you must use the command line.

To change the root device volume of an instance to persist at launch using the console

1. Open the Amazon EC2 console.
2. From the Amazon EC2 console dashboard, click **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, choose the AMI to use and click **Select**.
4. Follow the wizard to complete the **Choose an Instance Type** and **Configure Instance Details** pages.
5. On the **Add Storage** page, deselect the **Delete On Termination** check box for the root volume.
6. Complete the remaining wizard pages, and then click **Launch**.

You can verify the setting by viewing details for the root device volume on the instance's details pane. Next to **Block devices**, click the entry for the root device volume. By default, **Delete on termination** is `True`. If you change the default behavior, **Delete on termination** is `False`.

To change the root device volume of an instance to persist using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- modify-instance-attribute (AWS CLI)
- ec2-modify-instance-attribute (Amazon EC2 CLI)
- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

Networking and Security

You can launch instances in one of two platforms: EC2-Classic and EC2-VPC. An instance that's launched into EC2-Classic is assigned a public IP address. By default, an instance that's launched into EC2-VPC is assigned public IP address only if it's launched into a default VPC. An instance that's launched into a nondefault VPC must be specifically assigned a public IP address at launch, or you must modify your subnet's default public IP addressing behavior. For more information about EC2-Classic and EC2-VPC, see Supported Platforms (p. 455).

Instances can fail or terminate for reasons outside of your control. If one fails and you launch a replacement instance, the replacement has a different public IP address than the original. However, if your application needs a static IP address, Amazon EC2 offers *Elastic IP addresses*. For more information, see Amazon EC2 Instance IP Addressing (p. 474).

You can use *security groups* to control who can access your instances. These are analogous to an inbound network firewall that enables you to specify the protocols, ports, and source IP ranges that are allowed to reach your instances. You can create multiple security groups and assign different rules to each group. You can then assign each instance to one or more security groups, and we use the rules to determine which traffic is allowed to reach the instance. You can configure a security group so that only specific IP addresses or specific security groups have access to the instance. For more information, see Amazon EC2 Security Groups for Windows Instances (p. 398).

AWS Identity and Access Management

AWS Identity and Access Management (IAM) enables you to do the following:

- Create users and groups under your AWS account
- Assign unique security credentials to each user under your AWS account
- Control each user's permissions to perform tasks using AWS resources
- Allow the users in another AWS account to share your AWS resources
- Create roles for your AWS account and define the users or services that can assume them
- Use existing identities for your enterprise to grant permissions to perform tasks using AWS resources

By using IAM with Amazon EC2, you can control whether users in your organization can perform a task using specific Amazon EC2 API actions and whether they can use specific AWS resources.

For more information about IAM, see the following:

- Creating an IAM Group and Users (p. 407)
- IAM Policies for Amazon EC2 (p. 408)
- IAM Roles for Amazon EC2 (p. 442)
- Identity and Access Management (IAM)
- IAM User Guide

Differences between Windows Server and an Amazon EC2 Windows Instance

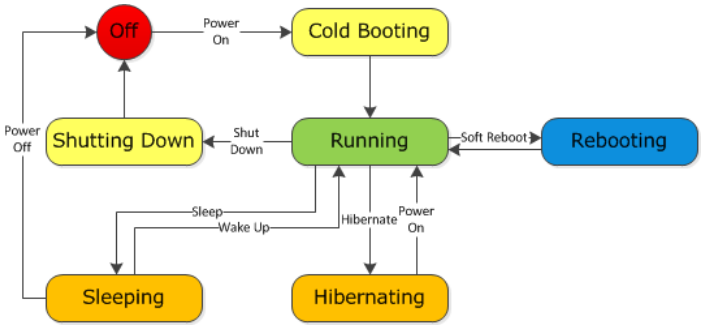
After you launch your Amazon EC2 Windows instance, it behaves like a traditional server running Windows Server. For example, both Windows Server and an Amazon EC2 instance can be used to run your web applications, conduct batch processing, or manage applications requiring large-scale computations. However, there are important differences between the server hardware model and the cloud computing model. The way an Amazon EC2 instance runs is not the same as the way a traditional server running Windows Server runs.

Before you begin launching Amazon EC2 Windows instances, you should be aware that the architecture of applications running on cloud servers can differ significantly from the architecture for traditional application models running on your hardware. Implementing applications on cloud servers requires a shift in your design process.

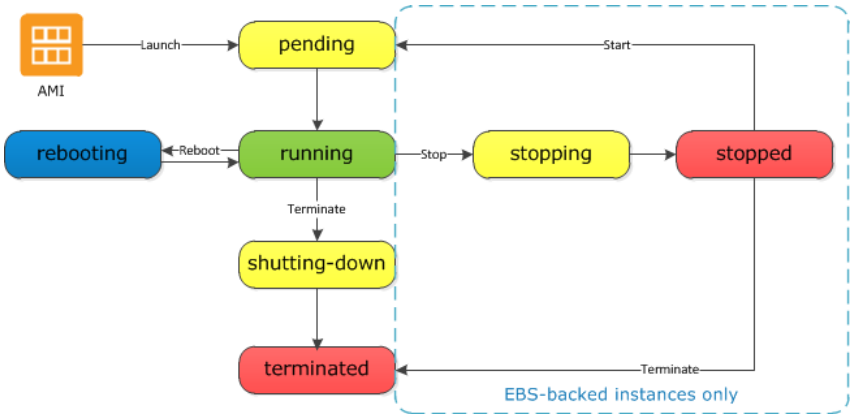
The following table describes some key differences between Windows Server and an Amazon EC2 Windows instance.

Windows Server	Amazon EC2 Windows Instance
Resources and capacity are physically limited.	Resources and capacity are scalable.
You pay for the infrastructure, even if you don't use it.	You pay for the usage of the infrastructure. We stop charging you for the instance as soon as you stop or terminate it.
Occupies physical space and must be maintained on a regular basis.	Doesn't occupy physical space and does not require regular maintenance.
Starts with push of the power button (known as *cold booting*).	Starts with the launch of the instance.
You can keep the server running until it is time to shut it down, or put it in a sleep or hibernation state (during which the server is powered down).	You can keep the server running, or stop and restart it (during which the instance is moved to a new host computer).
When you shut down the server, all resources remain intact and in the state they were in when you switched it off. Information you stored on the hard drives persists and can be accessed whenever it's needed. You can restore the server to the running state by powering it on.	When you terminate the instance, its infrastructure is no longer available to you. You can't connect to or restart an instance after you've terminated it. However, you can create an image from your instance while it's running, and launch new instances from the image at any time.

A traditional server running Windows Server goes through the states shown in the following diagram.



An Amazon EC2 Windows instance is similar to the traditional Windows Server, as you can see by comparing the following diagram with the previous diagram for Windows Server. After you launch an instance, it briefly goes into the pending state while registration takes place, then it goes into the running state. The instance remains active until you stop or terminate it. You can't restart an instance after you terminate it. You can create a backup image of your instance while it's running, and launch a new instance from that backup image.



Designing Your Applications to Run on Amazon EC2 Windows Instances

It is important that you consider the differences mentioned in the previous section when you design your applications to run on Amazon EC2 Windows instances.

Applications built for Amazon EC2 use the underlying computing infrastructure on an as-needed basis. They draw on necessary resources (such as storage and computing) on demand in order to perform a job, and relinquish the resources when done. In addition, they often dispose of themselves after the job is done. While in operation, the application scales up and down elastically based on resource requirements. An application running on an Amazon EC2 instance can terminate and recreate the various components at will in case of infrastructure failures.

When designing your Windows applications to run on Amazon EC2, you can plan for rapid deployment and rapid reduction of compute and storage resources, based on your changing needs.

When you run an Amazon EC2 Windows instance, you don't need to provision the exact system package of hardware, software, and storage, the way you do with Windows Server. Instead, you can focus on using a variety of cloud resources to improve the scalability and overall performance of your Windows application.

With Amazon EC2, designing for failure and outages is an integral and crucial part of the architecture. As with any scalable and redundant system, architecture of your system should account for computing, network, and storage failures. You have to build mechanisms in your applications that can handle different kinds of failures. The key is to build a modular system with individual components that are not tightly coupled, can interact asynchronously, and treat one another as black boxes that are independently scalable. Thus, if one of your components fails or is busy, you can launch more instances of that component without breaking your current system.

Another key element to designing for failure is to distribute your application geographically. Replicating your application across geographically distributed regions improves high availability in your system.

Amazon EC2 infrastructure is programmable and you can use scripts to automate the deployment process, to install and configure software and applications, and to bootstrap your virtual servers.

You should implement security in every layer of your application architecture running on an Amazon EC2 Windows instance. If you are concerned about storing sensitive and confidential data within your Amazon EC2 environment, you should encrypt the data before uploading it.

Setting Up with Amazon EC2

If you've already signed up for Amazon Web Services (AWS), you can start using Amazon EC2 immediately. You can open the Amazon EC2 console, click **Launch Instance**, and follow the steps in the launch wizard to launch your first instance.

If you haven't signed up for AWS yet, or if you need assistance launching your first instance, complete the following tasks to get set up to use Amazon EC2:

1. Sign Up for AWS (p. 14)
2. Create an IAM User (p. 15)
3. Create a Key Pair (p. 16)
4. Create a Virtual Private Cloud (VPC) (p. 17)
5. Create a Security Group (p. 17)

Sign Up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all services in AWS, including Amazon EC2. You are charged only for the services that you use.

With Amazon EC2, you pay only for what you use. If you are a new AWS customer, you can get started with Amazon EC2 for free. For more information, see AWS Free Tier.

If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

To create an AWS account

1. Open http://aws.amazon.com/, and then click **Sign Up**.
2. Follow the on-screen instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Note your AWS account number, because you'll need it for the next task.

Create an IAM User

Services in AWS, such as Amazon EC2, require that you provide credentials when you access them, so that the service can determine whether you have permission to access its resources. The console requires your password. You can create access keys for your AWS account to access the command line interface or API. However, we don't recommend that you access AWS using the credentials for your AWS account; we recommend that you use AWS Identity and Access Management (IAM) instead. Create an IAM user, and then add the user to an IAM group with administrative permissions or and grant this user administrative permissions. You can then access AWS using a special URL and the credentials for the IAM user.

If you signed up for AWS but have not created an IAM user for yourself, you can create one using the IAM console. If you aren't familiar with using the console, see Working with the AWS Management Console for an overview.

To create the Administrators group

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, click **Groups**, and then click **Create New Group**.
3. In the **Group Name** box, type `Administrators` , and then click **Next Step**.
4. In the list of policies, select the check box next to the **AdministratorAccess** policy. You can use the **Filter** menu and the **Search** box to filter the list of policies.
5. Click **Next Step**, and then click **Create Group**.

Your new group is listed under **Group Name**.

To create an IAM user for yourself, add the user to the Administrators group, and create a password for the user

1. In the navigation pane, click **Users**, and then click **Create New Users**.
2. In box **1**, type a user name. Clear the check box next to **Generate an access key for each user**. Then click **Create**.
3. In the list of users, click the name (not the check box) of the user you just created. You can use the **Search** box to search for the user name.
4. In the **Groups** section, click **Add User to Groups**.
5. Select the check box next to the **Administrators** group. Then click **Add to Groups**.
6. Scroll down to the **Security Credentials** section. Under **Sign-In Credentials**, click **Manage Password**.
7. Select **Assign a custom password**. Then type a password in the **Password** and **Confirm Password** boxes. When you are finished, click **Apply** .

To sign in as this new IAM user, sign out of the AWS console, then use the following URL, where *your_aws_account_id* is your AWS account number without the hyphens (for example, if your AWS account number is `1234-5678-9012`, your AWS account ID is `123456789012`):

```
https://your_aws_account_id.signin.aws.amazon.com/console/
```

Enter the IAM user name and password that you just created. When you're signed in, the navigation bar displays "*your_user_name* @ *your_aws_account_id*".

If you don't want the URL for your sign-in page to contain your AWS account ID, you can create an account alias. From the IAM dashboard, click **Customize** and enter an alias, such as your company name. To sign in after you create an account alias, use the following URL:

```
https://your_account_alias.signin.aws.amazon.com/console/
```

To verify the sign-in link for IAM users for your account, open the IAM console and check under **IAM users sign-in link** on the dashboard.

For more information about IAM, see IAM and Amazon EC2 (p. 407).

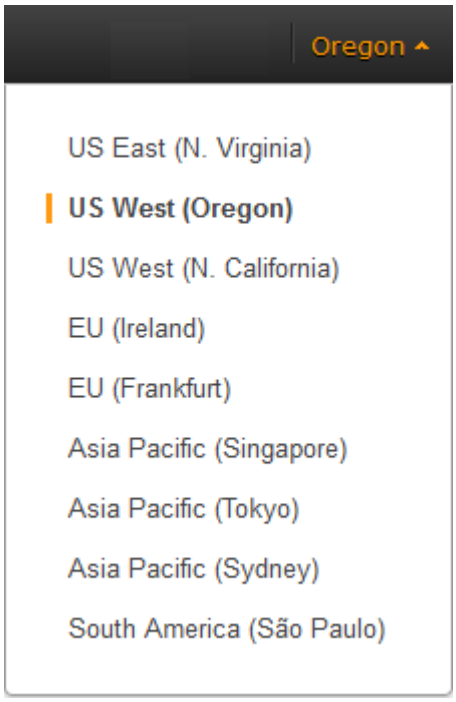
Create a Key Pair

AWS uses public-key cryptography to secure the login information for your instance. You specify the name of the key pair when you launch your instance, then provide the private key to obtain the administrator password for your Windows instance so you can log in using RDP.

If you haven't created a key pair already, you can create one using the Amazon EC2 console. Note that if you plan to launch instances in multiple regions, you'll need to create a key pair in each region. For more information about regions, see Regions and Availability Zones (p. 6).

To create a key pair

1. Sign in to AWS using the URL that you created in the previous section. Open the Amazon EC2 console.
2. From the navigation bar, select a region for the key pair. You can select any region that's available to you, regardless of your location. However, key pairs are specific to a region; for example, if you plan to launch an instance in the US West (Oregon) region, you must create a key pair for the instance in the US West (Oregon) region.



3. Click **Key Pairs** in the navigation pane.
4. Click **Create Key Pair**.
5. Enter a name for the new key pair in the **Key pair name** field of the **Create Key Pair** dialog box, and then click **Create**. Choose a name that is easy for you to remember, such as your IAM user name, followed by `-key-pair`, plus the region name. For example, *me*-key-pair-*uswest2*.

6. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is `.pem`. Save the private key file in a safe place.

   Important
   This is the only chance for you to save the private key file. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

For more information, see Amazon EC2 Key Pairs and Windows Instances (p. 394).

Create a Virtual Private Cloud (VPC)

Amazon VPC enables you to launch AWS resources into a virtual network that you've defined. If you have a default VPC, you can skip this section and move to the next task, Create a Security Group (p. 17). To determine whether you have a default VPC, see Supported Platforms in the Amazon EC2 Console (p. 456). Otherwise, you can create a nondefault VPC in your account using the steps below.

Important
If your account supports EC2-Classic in a region, then you do not have a default VPC in that region. T2 instances must be launched into a VPC.

To create a nondefault VPC

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. From the navigation bar, select a region for the VPC. VPCs are specific to a region, so you should select the same region in which you created your key pair.
3. On the VPC dashboard, click **Start VPC Wizard**.
4. On the **Step 1: Select a VPC Configuration** page, ensure that **VPC with a Single Public Subnet** is selected, and click **Select**.
5. On the **Step 2: VPC with a Single Public Subnet** page, enter a friendly name for your VPC in the **VPC name** field. Leave the other default configuration settings, and click **Create VPC**. On the confirmation page, click **OK**.

For more information about Amazon VPC, see What is Amazon VPC? in the *Amazon VPC User Guide*.

Create a Security Group

Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level. You must add rules to a security group that enable you to connect to your instance from your IP address using RDP. You can also add rules that allow inbound and outbound HTTP and HTTPS access from anywhere.

Note that if you plan to launch instances in multiple regions, you'll need to create a security group in each region. For more information about regions, see Regions and Availability Zones (p. 6).

Prerequisites

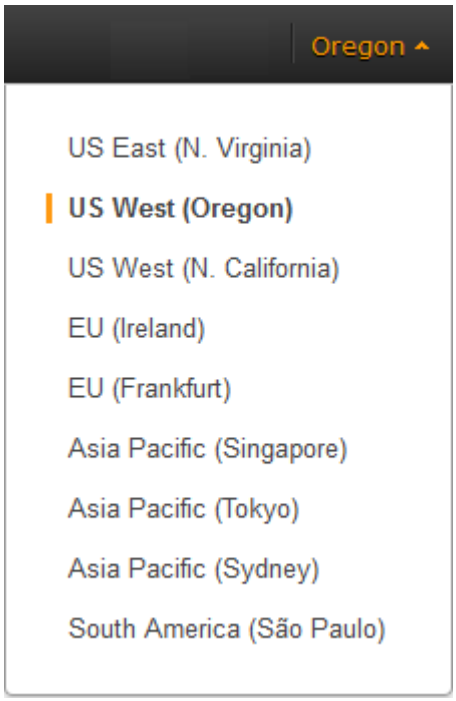
You'll need the public IP address of your local computer, which you can get using a service. For example, we provide the following service: http://checkip.amazonaws.com/. To locate another service that provides your IP address, use the search phrase "what is my IP address." If you are connecting through an Internet service provider (ISP) or from behind a firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

To create a security group with least privilege

1. Open the Amazon EC2 console.

   Tip
   Alternatively, you can use the Amazon VPC console to create a security group. However, the instructions in this procedure don't match the Amazon VPC console. Therefore, if you switched to the Amazon VPC console in the previous section, either switch back to the Amazon EC2 console and use these instructions, or use the instructions in Set Up a Security Group for Your VPC in the *Amazon VPC Getting Started Guide*.

2. From the navigation bar, select a region for the security group. Security groups are specific to a region, so you should select the same region in which you created your key pair.



3. Click **Security Groups** in the navigation pane.
4. Click **Create Security Group**.
5. Enter a name for the new security group and a description. Choose a name that is easy for you to remember, such as your IAM user name, followed by _SG_, plus the region name. For example, *me*_SG_*uswest2*.
6. In the **VPC** list, select your VPC. If you have a default VPC, it's the one that is marked with an asterisk (*).

   Note
   If your account supports EC2-Classic, select the VPC that you created in the previous task.

7. On the **Inbound** tab, create the following rules (click **Add Rule** for each new rule), and then click **Create**:

   - Select **HTTP** from the **Type** list, and make sure that **Source** is set to **Anywhere** (`0.0.0.0/0`).
   - Select **HTTPS** from the **Type** list, and make sure that **Source** is set to **Anywhere** (`0.0.0.0/0`).
   - Select **RDP** from the **Type** list. In the **Source** box, ensure **Custom IP** is selected, and specify the public IP address of your computer or network in CIDR notation. To specify an individual IP address in CIDR notation, add the routing prefix `/32`. For example, if your IP address is `203.0.113.25`, specify `203.0.113.25/32`. If your company allocates addresses from a range, specify the entire range, such as `203.0.113.0/24`.

Caution
For security reasons, we don't recommend that you allow RDP access from all IP addresses (`0.0.0.0/0`) to your instance, except for testing purposes and only for a short time.

For more information, see Amazon EC2 Security Groups for Windows Instances (p. 398).

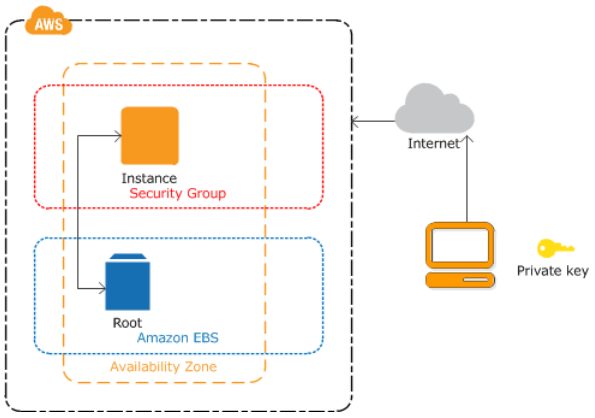
Getting Started with Amazon EC2 Windows Instances

This tutorial provides a hands-on introduction to using Amazon EC2 using the AWS Management Console, a point-and-click web-based interface. We'll launch and connect to a Windows instance.

Important
Before you begin, be sure that you've completed the steps in Setting Up with Amazon EC2 (p. 14).

Overview

The instance is an Amazon EBS-backed instance (meaning that the root volume is an Amazon EBS volume) running Windows Server. You can either specify the Availability Zone in which your instance runs, or let us select an Availability Zone for you. When you launch your instance, you secure it by specifying a key pair and a security group. When you connect to your instance, you must specify the private key of the key pair that you specified when launching your instance. Your instance looks like a traditional host, and you can interact with it as you would any computer running Windows Server.



To complete this tutorial

1. Launch a Windows Instance (p. 21)
2. Connecting to Your Windows Instance Using RDP (p. 216)

3. (Optional) Create a CloudWatch Alarm to Monitor Your Instance (p. 24).
4. Clean Up (p. 26)

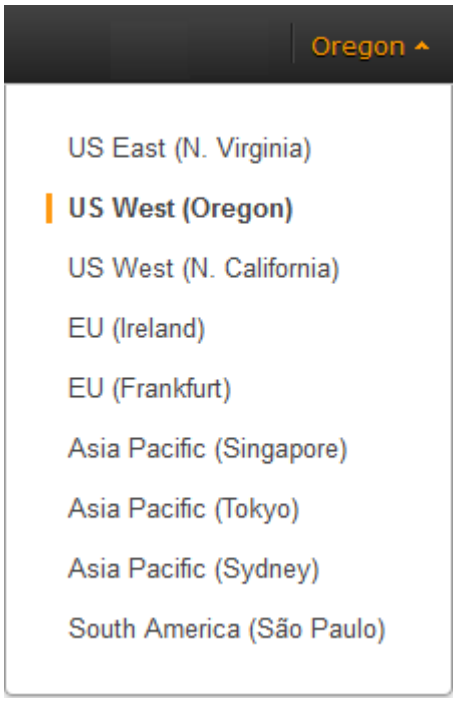
If you'd prefer to launch a Linux instance, see this tutorial in the *Amazon EC2 User Guide for Linux Instances*: Getting Started with Amazon EC2 Linux Instances.

Launch a Windows Instance

You can launch a Windows instance using the AWS Management Console as described in the following procedure. An instance is a virtual server in the AWS cloud. With Amazon EC2, you can set up and configure the operating system and applications that run on your instance.

To launch an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the region for the instance. For this tutorial, you can use the default region. Otherwise, this choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For example, if you'd like to connect your instance to an existing Amazon EBS volume, you must select the same region as the volume.



3. On the console dashboard, choose **Launch Instance**.
4. The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations, called *Amazon Machine Images (AMIs)*, that serve as templates for your instance. Select the 64-bit version of Microsoft Windows Server 2008 R2. Notice that this configuration is marked "Free tier eligible."
5. On the **Choose an Instance Type** page, you can select the hardware configuration of your instance. Select one of the following instance types, which are the only instance types eligible for the free tier.

   - `t2.micro` (which is selected by default)
   - `t1.micro` (select `All generations` from the filter list instead of `Current generation`, and then select `t1.micro`)

6. (`t2.micro`) T2 instances, such as `t2.micro`, must be launched into a VPC. If your AWS account supports EC2-Classic and you do not have a VPC in the selected region, the launch wizard creates a VPC for you and you can continue to the next step in this procedure. Otherwise, if you have one or more VPCs (such as a default VPC), the **Review and Launch** button is disabled and you must do the following:

   a. Choose **Next: Configure Instance Details**.
   b. Select your VPC from the **Network** list and select a subnet from the **Subnet** list.
   c. Select `Enable` from **Auto-assign Public IP**. Note that `Enable` is the default only if the VPC is a default VPC.

7. Choose **Review and Launch** to let the wizard complete the other configuration settings for you.
8. On the **Review Instance Launch** page, under **Security Groups**, you'll see that the wizard created and selected a security group for you. Instead, select the security group that you created when getting set up using the following steps:

   a. Choose **Edit security groups**.
   b. On the **Configure Security Group** page, ensure the **Select an existing security group** option is selected.
   c. Select your security group from the list of existing security groups, and choose **Review and Launch**.

9. On the **Review Instance Launch** page, choose **Launch**.
10. In the **Select an existing key pair or create a new key pair** dialog box, you can select **Choose an existing key pair**, to select a key pair you already created.

    Alternatively, you can create a new key pair. Select **Create a new key pair**, enter a name for the key pair, and then choose **Download Key Pair**. This is the only chance for you to save the private key file, so be sure to download it. Save the private key file in a safe place. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

    Caution
    Don't select the **Proceed without a key pair** option. If you launch your instance without a key pair, then you can't connect to it.

    When you are ready, select the acknowledgement check box, and then choose **Launch Instances**.
11. A confirmation page lets you know that your instance is launching. Choose **View Instances** to close the confirmation page and return to the console.
12. On the **Instances** page, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is **pending**. After the instance starts, its state changes to **running** and it receives a public DNS name. (If the **Public DNS** column is hidden, choose the Show/Hide icon in the top right corner of the **Instances** page and select **Public DNS**.)
13. Record the public DNS name for your instance because you'll need it for the next step.
14. (Optional) After your instance is launched, you can view its security group rules. From the Instances page, select the instance. In the **Description** tab, find **Security groups** and choose **view rules**.

As you can see, if you used the security group the wizard created for you, it contains one rule that allows RDP traffic from any IP source to port 3389. If you launch a Windows instance running IIS and SQL, the wizard creates a security group that contains additional rules to allow traffic to port 80 for HTTP (for IIS) and port 1433 for MS SQL.

Connect to Your Windows Instance

To connect to a Windows instance, you must retrieve the initial administrator password and then specify this password when you connect to your instance using Remote Desktop.

Note
If you've joined your instance to a domain, you can connect to your instance using domain credentials you've defined in AWS Directory Service. For more information about connecting to an instance in a domain, see Connecting To Your Instance Using Domain Credentials (p. 297).

The name of the administrator account depends on the language of the operating system. For example, for English, it's Administrator, for French it's Administrateur, and for Portuguese it's Administrador. For more information, see Localized Names for Administrator Account in Windows in the Microsoft TechNet Wiki.

Windows instances are limited to two simultaneous remote connections at one time. If you attempt a third connection, an error will occur. For more information, see Configure the Number of Simultaneous Remote Connections Allowed for a Connection.

To connect to your Windows instance using an RDP client

1. In the Amazon EC2 console, select the instance, and then choose **Connect**.
2. In the **Connect To Your Instance** dialog box, choose **Get Password** (it will take a few minutes after the instance is launched before the password is available).
3. Choose **Browse** and navigate to the private key file you created when you launched the instance. Select the file and choose **Open** to copy the entire contents of the file into contents box.
4. Choose **Decrypt Password**. The console displays the default administrator password for the instance in the **Connect To Your Instance** dialog box, replacing the link to **Get Password** shown previously with the actual password.
5. Record the default administrator password, or copy it to the clipboard. You need this password to connect to the instance.
6. Choose **Download Remote Desktop File**. Your browser prompts you to either open or save the .rdp file. Either option is fine. When you have finished, you can choose **Close** to dismiss the **Connect To Your Instance** dialog box.

   - If you opened the .rdp file, you'll see the **Remote Desktop Connection** dialog box.
   - If you saved the .rdp file, navigate to your downloads directory, and open the .rdp file to display the dialog box.

7. You may get a warning that the publisher of the remote connection is unknown. If you are using **Remote Desktop Connection** from a Windows PC, choose **Connect** to connect to your instance. If you are using **Microsoft Remote Desktop** on a Mac, skip the next step.
8. When prompted, log in to the instance, using the administrator account for the operating system and the password that you recorded or copied previously. If your **Remote Desktop Connection** already has an administrator account set up, you might have to choose the **Use another account** option and enter the user name and password manually.

   Note
   Sometimes copying and pasting content can corrupt data. If you encounter a "Password Failed" error when you log in, try typing in the password manually.

9. Due to the nature of self-signed certificates, you may get a warning that the security certificate could not be authenticated. Use the following steps to verify the identity of the remote computer, or simply choose **Yes** or **Continue** to continue if you trust the certificate.

    a. If you are using **Remote Desktop Connection** from a Windows PC, choose **View certificate**. If you are using **Microsoft Remote Desktop** on a Mac, choose **Show Certificate**.

    b. Choose the **Details** tab, and scroll down to the **Thumbprint** entry on a Windows PC, or the **SHA1 Fingerprints** entry on a Mac. This is the unique identifier for the remote computer's security certificate.

    c. In the Amazon EC2 console, select the instance, choose **Actions**, and then choose **Get System Log**.

    d. In the system log output, look for an entry labeled `RDPCERTIFICATE-THUMBPRINT`. If this value matches the thumbprint or fingerprint of the certificate, you have verified the identity of the remote computer.

    e. If you are using **Remote Desktop Connection** from a Windows PC, return to the **Certificate** dialog box and choose **OK**. If you are using **Microsoft Remote Desktop** on a Mac, return to the **Verify Certificate** and choose **Continue**.

    f. If you are using **Remote Desktop Connection** from a Windows PC, choose **Yes** in the **Remote Desktop Connection** window to connect to your instance. If you are using **Microsoft Remote Desktop** on a Mac, log in to the instance as prompted, using the default **Administrator** account and the default administrator password that you recorded or copied previously.

        Note
        On a Mac, you may need to switch spaces to see the **Microsoft Remote Desktop** login screen. For more information on spaces, see http://support.apple.com/kb/PH14155.

Create a CloudWatch Alarm to Monitor Your Instance

With Amazon CloudWatch, you can monitor various aspects of your instance and set up alarms based on criteria you choose. For example, you could configure an alarm to send you an email when an instance's CPU exceeds 70 percent.

Because you just launched your instance, it is unlikely that the CPU will exceed this threshold, so instead, set a CloudWatch alarm to send you an email when your instance's CPU is *lower than* 70 percent for five minutes. For more information about CloudWatch see What is Amazon CloudWatch in the *Amazon CloudWatch Developer Guide*.

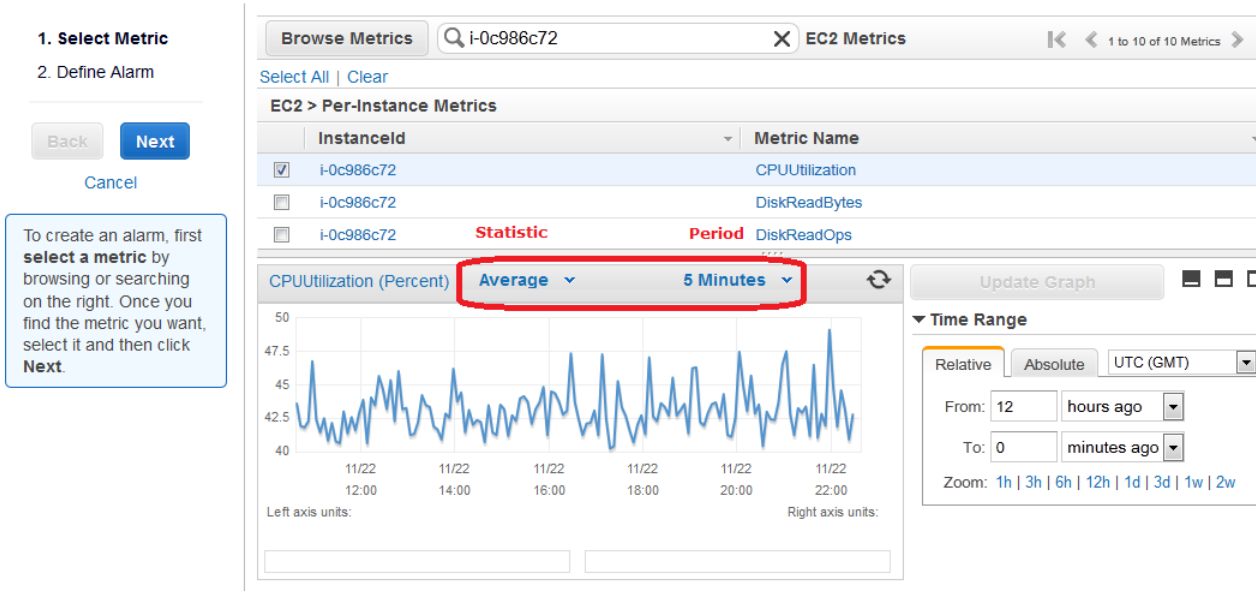
To create an alarm to monitor your instance

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region to match the region in which you launched the instance.
3. In the navigation pane, choose **Alarms**.
4. Choose **Create Alarm**, and then in the **CloudWatch Metrics by Category** pane, select **EC2 Metrics**.
5. Select a metric using the following procedure, and then choose **Next**:

    a. In the list of metrics, select the row that contains `CPUUtilization` for your instance.

    b. Select **`Average`** from the statistic drop-down list.

    c. Select a period from the period drop-down list, for example: **`5 Minutes`**.



6. Define the alarm using the following procedure, and then choose **Create Alarm**:

   a. Under **Alarm Threshold**, in the **Name** box, enter a unique name for the alarm, for example: **`myTestAlarm`**.

   b. In the **Description** field, enter a description of the alarm, for example: **`CPU usage is lower than 70 percent.`**

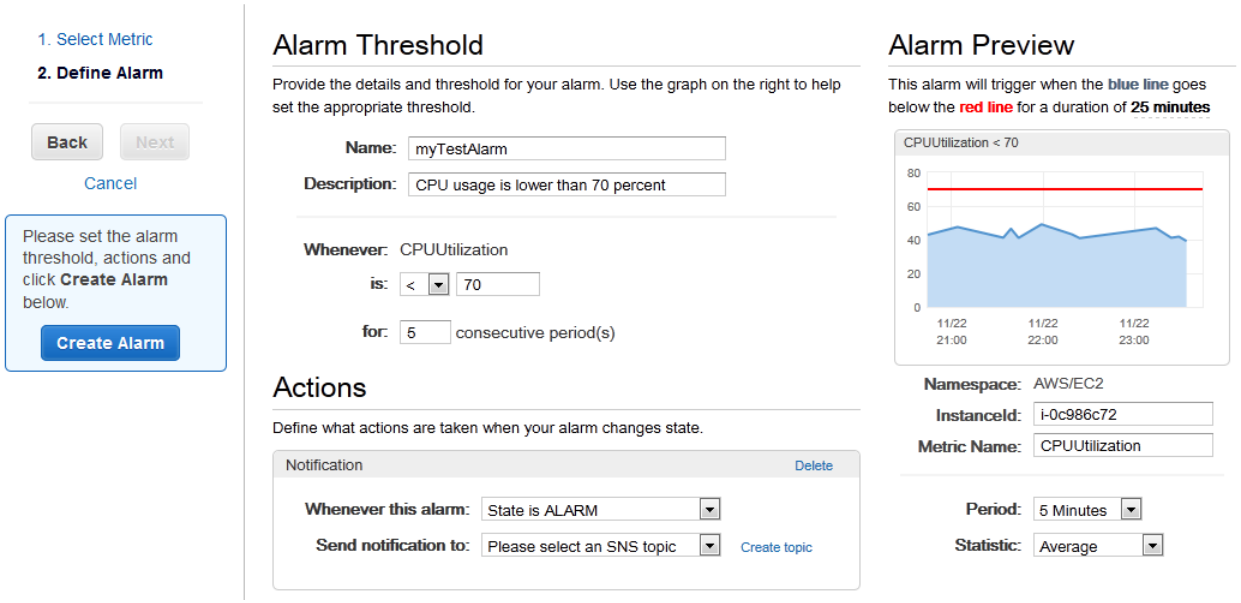
   c. Under **Whenever**, next to **is**, select **<** from the list and enter **`70`** in the box.

   d. Under **Whenever**, next to **for**, enter **`5`** in the box.

      We display a graphical representation of the threshold under **Alarm Preview**.

   e. Under **Actions**, in the **Whenever this alarm** drop-down list, select **State is ALARM**.

   f. In the **Send notification to** list, select an existing Amazon SNS topic or create a new one. To create a new Amazon SNS topic, choose **Create topic**. In **Send notification to**, enter a name for the new Amazon SNS topic. In **Email list**, enter a comma-separated list of email addresses.



7. We'll send a notification email to the email address you specified with a link to an opt-in confirmation page for your notification. After you opt in, we'll send a notification email when the instance has been running for more than 5 minutes at less than 70 percent CPU utilization.

Clean Up

Now that you've completed this tutorial, you can clean up the resources that you created. You could also customize your instance to your needs and keep using it.

> **Important**
> Remember, unless you are within the AWS Free Tier, as soon as your instance starts to boot, you're billed for each hour or partial hour that you keep the instance running (even if the instance is idle).

When you've decided that you no longer need the instance, you need to clean up these resources:

- The Amazon CloudWatch alarm
- The instance

To delete your CloudWatch alarm

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Alarms**.
3. In the alarms list, select the alarm you created, and then choose **Delete**.

Terminating an instance effectively deletes it; you can't reconnect to an instance after you've terminated it.

If you launched an instance that is not within the AWS Free Tier, you'll stop incurring charges for that instance as soon as the instance status changes to `shutting down` or `terminated`.

To terminate your instance

1. In the navigation pane, choose **Instances**. In the list of instances, select the instance you want to terminate.
2. Choose **Actions**, then **Instance State**, and then choose **Terminate**.
3. Choose **Yes, Terminate** when prompted for confirmation.

Best Practices for Amazon EC2

This checklist is intended to help you get the maximum benefit from and satisfaction with Amazon EC2.

Security and Network

- Manage access to AWS resources and APIs using identity federation, IAM users, and IAM roles. Establish credential management policies and procedures for creating, distributing, rotating, and revoking AWS access credentials. For more information, see IAM Best Practices in the *IAM User Guide*.
- Implement the least permissive rules for your security group. For more information, see Security Group Rules (p. 400).
- Regularly patch, update, and secure the operating system and applications on your instance. For more information about updating Amazon Linux, see Managing Software on Your Linux Instance in the *Amazon EC2 User Guide for Linux Instances*. For more information about updating your Windows instance, see Updating Your Windows Instance.
- Launch your instances into a VPC instead of EC2-Classic. Note that if you created your AWS account after 2013-12-04, we automatically launch your instances into a VPC. For more information about the benefits, see Amazon EC2 and Amazon Virtual Private Cloud (p. 449).

Storage

- Understand the implications of the root device type for data persistence, backup, and recovery. For more information, see Storage for the Root Device (p. 53).
- Use separate Amazon EBS volumes for the operating system versus your data. Ensure that the volume with your data persists after instance termination. For more information, see Preserving Amazon EBS Volumes on Instance Termination (p. 227).
- Use the instance store available for your instance to store temporary data. Remember that the data stored in instance store is deleted when you stop or terminate your instance. If you use instance store for database storage, ensure that you have a cluster with a replication factor that ensures fault tolerance.

Resource Management

- Use instance metadata and custom resource tags to track and identify your AWS resources. For more information, see Instance Metadata and User Data (p. 160) and Tagging Your Amazon EC2 Resources (p. 609).
- View your current limits for Amazon EC2. Plan to request any limit increases in advance of the time that you'll need them. For more information, see Amazon EC2 Service Limits (p. 618).

Backup and Recovery

- Regularly back up your instance using Amazon EBS snapshots (p. 549) or a backup tool.
- Deploy critical components of your application across multiple Availability Zones, and replicate your data appropriately.
- Design your applications to handle dynamic IP addressing when your instance restarts. For more information, see Amazon EC2 Instance IP Addressing (p. 474).
- Monitor and respond to events. For more information, see Monitoring Amazon EC2 (p. 318).
- Ensure that you are prepared to handle failover. For a basic solution, you can manually attach a network interface or Elastic IP address to a replacement instance. For more information, see Elastic Network Interfaces (ENI) (p. 492). For an automated solution, you can use Auto Scaling. For more information, see the Auto Scaling Developer Guide.
- Regularly test the process of recovering your instances and Amazon EBS volumes if they fail.

Tutorials for Amazon EC2 Instances Running Windows Server

The following tutorials show you how to perform commonly requested tasks with Amazon EC2 instances running Windows Server.

Tutorials

- Tutorial: Deploying a WordPress Blog on Your Amazon EC2 Instance Running Windows Server (p. 30)
- Tutorial: Installing a WAMP Server on an Amazon EC2 Instance Running Windows Server (p. 34)
- Tutorial: Installing a WIMP Server on an Amazon EC2 Instance Running Windows Server (p. 37)
- Tutorial: Setting Up a Windows HPC Cluster on Amazon EC2 (p. 41)

Tutorial: Deploying a WordPress Blog on Your Amazon EC2 Instance Running Windows Server

This tutorial will help you install and deploy a WordPress blog on an Amazon EC2 instance running Microsoft Windows Server.

If you'd prefer to host your WordPress blog on a Linux instance, see Tutorial: Hosting a WordPress Blog with Amazon EC2 in the *Amazon EC2 User Guide for Linux Instances*.

Prerequisites

Before you get started, be sure that you do the following:

1. Launch an Amazon EC2 instance from the Microsoft Windows Server 2008 R2 base AMI. For information about launching an instance, see Getting Started with Amazon EC2 Windows Instances (p. 20).
2. Use the AWS free usage tier (if eligible) to launch and use the free Windows *t2.micro* instance for 12 months. You can use the AWS free usage tier for launching new applications, testing existing applications, or simply gaining hands-on experience with AWS. For more information about eligibility and the highlights, see the AWS Free Usage Tier product page.

Important
If you've launched a regular instance and use it to deploy the WordPress website, you will incur the standard Amazon EC2 usage fees for the instance until you terminate it. For more information about Amazon EC2 usage rates, go to the Amazon EC2 product page.

3. Ensure that the security group in which you're launching your instance has ports 80 (HTTP), 443 (HTTPS), and 3389 (RDP) open for inbound traffic. Ports 80 and 443 allow computers outside of the instance to connect with HTTP and HTTPS. If these ports are not open, the WordPress site can't be accessed from outside the instance. Port 3389 allows you to connect to the instance with Remote Desktop Protocol.
4. Connect to your instance.

Installing the Microsoft Web Platform Installer

You can use the Microsoft Web Platform Installer to install and configure WordPress on your server. This tool simplifies deployment of Web applications and Web sites to IIS servers. For more information, see Microsoft Web Platform Installer.

1. Verify that you've met the conditions in Prerequisites (p. 30).
2. Disable Internet Explorer Enhanced Security Configuration.
   a. In your Windows instance, click **Start**, point to **Administrative Tools**, and then click **Server Manager**.
   b. Click **Server Manager** in the navigation pane on the left, look for **Configure IE ESC** in the **Security Information** section of the main pane on the right. Click **Configure IE ESC**.
   c. Under **Administrators**, click **Off** and click **OK**.
   d. Close the **Server Manager** window.
3. In the Windows instance, download and install the latest version of the Microsoft Web Platform Installer.
   a. Click **Start**, point to **All Programs**, and click **Internet Explorer**.
   b. Click **Yes** in the pop-up window to accept the recommended security settings for Internet Explorer.
   c. Paste the following URL into the Internet Explorer address bar: `http://www.microsoft.com/web/downloads/platform.aspx`
   d. Click the **Free Download** button on the Microsoft Web Platform Installer page to download the installer and then click **Run** to run the installer.

Installing WordPress

Now that the Web Platform Installer is installed, you can use it install and configure WordPress on your server.

To install WordPress

1. Open the **Web Platform Installer** and click **Applications**.
2. Select **WordPress**, click **Add**, and then click **Install**.
3. On the **Prerequisites** page, select **MySQL** for the database to use. Enter the desired administrator password for your MySQL database in the **Password** and **Re-type Password** boxes, and then click **Continue**.

Note
For more information about creating a secure password, see http://www.pctools.com/guides/password/. Do not reuse an existing password, and make sure to store this password in a safe place.

4. Click **I Accept** for the list of third-party application software, Microsoft products (including the IIS web server), and components. After the Web Platform Installer finishes installing the software, you are prompted to configure your new site.
5. On the **Configure** page, clear the default application name in the **'WordPress' application name:** box and leave it blank, then leave the default information in the other boxes and click **Continue**.
6. Click **Yes** to accept that the contents of the folder will be overwritten.

Configure Security Keys

WordPress allows you to generate and enter unique authentication keys and salts for your site. These key and salt values provide a layer of encryption to the browser cookies that WordPress users store on their local machines. Basically, adding long, random values here makes your site more secure.

For more information about security keys, see http://codex.wordpress.org/Editing_wp-config.php#Security_Keys.

To configure security keys

1. Visit https://api.wordpress.org/secret-key/1.1/salt/ to randomly generate a set of key values that you can copy and paste into the installation wizard. The following steps will show you how to modify these values in Notepad to work with a Windows installation.
2. Copy all of the text in that page to your clipboard. It should look similar to the example below.

   Note
   The values below are for example purposes only; do not use these values for your installation.

```
define('AUTH_KEY',         '3#U$$+[RXN8:b^-L 0(WU_+ c+WFkI~c]o]-
bHw+)/Aj[wTwSiZ<Qb[mghEXcRh-');
define('SECURE_AUTH_KEY',  'Zsz._P=l/|y.Lq)XjlkwS1y5NJ76E6EJ.AV0pCKZZB,*~*r
 ?6OP$eJT@;+(ndLg');
define('LOGGED_IN_KEY',    'ju}qwre3V*+8f_zOWf?{LlGsQ]Ye@2Jh^,8x>)Y
|;(^[Iw]Pi+LG#A4R?7N`YB3');
define('NONCE_KEY',
'P(g62HeZxEes|LnI^i=H,[XwK9I&[2s|:?0N}VJM%?;v2v]v+;+^9eXUahg@::Cj');
define('AUTH_SALT',        'C$DpB4Hj[JK:?{ql`sRVa:{:7yShy(9A@5wg+`JJVb1fk%_-
Bx*M4(qc[Qg%JT!h');
define('SECURE_AUTH_SALT', 'd!uRu#}+q#{f$Z?Z9uFPG.${+S{n~1M&%@~gL>U>NV<zpD-
@2-Es7Q1O-bp28EKv');
define('LOGGED_IN_SALT',   ';j{00P*owZf)kVD+FVLn-~
>.|Y%Ug4#I^*LVd9QeZ^&XmK|e(76miC+&W&+^0P/');
define('NONCE_SALT',
'-97r*V/cgxLmp?Zy4zUU4r99QQ_rGs2LTd%P;|_e1tS)8_B/,.6[=UK<J_y9?JWG');
```

3. Open a Notepad window by clicking **Start**, **All Programs**, **Accessories**, and then **Notepad**.
4. Paste the copied text into the Notepad window.
5. Windows WordPress installations do not accept the dollar sign ($) in key and salt values, so they need to be replaced with another character (such as S). In the Notepad window, click **Edit**, then click **Replace**.
6. In the **Find what** box, type $.
7. In the **Replace with** box, type S.

8. Click **Replace All** to replace all of the dollar signs with `S` characters.
9. Close the **Replace** window.
10. Paste the modified key and salt values from the Notepad window into their corresponding boxes in the installation wizard. For example, the `AUTH_KEY` value in the Notepad window should be pasted into the **Authentication Key** box in the wizard.

    Do not include the single quotes or other text surrounding the values, just the actual value as in the example shown below.

    The modified `AUTH_KEY` line from the Notepad window:

    ```
    define('AUTH_KEY',         '3#USS+[RXN8:b^-L 0(WU_+ c+WFkI~c]o]-
    bHw+)/Aj[wTwSiZ<Qb[mghEXcRh-');
    ```

    Paste this text into the **Authentication Key** box of the wizard:

    ```
    3#USS+[RXN8:b^-L 0(WU_+ c+WFkI~c]o]-bHw+)/Aj[wTwSiZ<Qb[mghEXcRh-
    ```

11. Click **Continue** and **Finish** to complete the Web Platform Installer wizard.

Administrative Information

When you complete the Web Platform Installer wizard, a browser window opens to your WordPress installation at `http://localhost/wp-admin/install.php`. On this page, you configure the title for your site and an administrative user to moderate your blog.

To complete the installation

1. On the WordPress **Welcome** page, enter the following information and click **Install WordPress**.

Field	Value
Site Title	Enter a name for your WordPress site.
Username	Enter a name for your WordPress administrator. For security purposes you should choose a unique name for this user, since this will be more difficult to exploit than the default user name, `admin`.
Password	Enter a strong password, and then enter it again to confirm. Do not reuse an existing password, and make sure to store this password in a safe place.
Your E-mail	Enter the email address you want to use for notifications.
Privacy	Check to allow search engines to index your site.

2. Click **Log In**.
3. On the **Log In** page, enter your user name for **Username** and the site password you entered previously for **Password**.

Making Your WordPress Site Public

Now that you can see your WordPress blog on your local host, you can publish this website as the default site on your instance so that other people can see it. The next procedure walks you through the process of modifying your WordPress settings to point to the public DNS name of your instance instead of your local host.

To configure the default settings for your WordPress site

1. Open the WordPress dashboard by opening a browser on your instance and going to `http://localhost/wp-admin`. If prompted for your credentials, enter your user name for the **Username** and your site password for **Password**.
2. In the **Dashboard** pane, click **Settings**.
3. On the **General Settings** page, enter the following information and click **Save Changes**.

   - **WordPress address (URL)**—The public DNS address of your instance. For example, your URL may look something like `http://ec2-203-0-113-25.compute-1.amazonaws.com`.

     You can get the public DNS for your instance using the Amazon EC2 console (select the instance and check the **Public DNS** column; if this column is hidden, click the **Show/Hide** icon and select **Public DNS**).
   - **Site address (URL)**—The same public DNS address of your instance that you set in **WordPress address (URL)**.

4. To see your new site, open a browser on a computer other than the instance hosting WordPress and type the public DNS address of your instance in the web address field. Your WordPress site appears.

Congratulations! You have just deployed a WordPress site on a Windows instance. If you no longer need this instance, you can remove it to avoid incurring charges. See Clean Up (p. 26) for instructions.

If your WordPress blog becomes popular and you need more compute power, you might consider migrating to a larger instance type; for more information, see Resizing Your Instance (p. 118). If your blog requires more storage space than you originally accounted for, you could expand the storage space on your instance (see Expanding the Storage Space of an EBS Volume on Windows (p. 544)). If your MySQL database needs to grow, you could consider moving it to Amazon RDS to take advantage of the service's autoscaling abilities.

For information about WordPress, see the WordPress Codex help documentation at http://codex.wordpress.org/. For more information about troubleshooting your installation, see http://codex.wordpress.org/Installing_WordPress#Common_Installation_Problems. For information about making your WordPress blog more secure, see http://codex.wordpress.org/Hardening_WordPress. For information about keeping your WordPress blog up-to-date, see http://codex.wordpress.org/Updating_WordPress.

Tutorial: Installing a WAMP Server on an Amazon EC2 Instance Running Windows Server

This tutorial shows you how to install an Apache web server with PHP and MySQL on an EC2 instance running Microsoft Windows Server. This software configuration is sometimes called a WAMP server or WAMP stack (Windows, Apache, MySQL, PHP). A WAMP stack is not designed for production environments because MySQL and Apache would compete for server resources. You can, however, create a WAMP stack on an EC2 instance to prototype a web project in a controlled test environment.

For example, you can host a static website or deploy a dynamic PHP application that reads and writes information to a database. For information about how to create a similar server on Linux, see Tutorial: Installing a LAMP Web Server in the *Amazon EC2 User Guide for Linux Instances*.

> **Note**
> There are many third-party solutions that you can use to install a WAMP stack; this tutorial uses the Bitnami WAMP stack. For more information, see Review: WAMP stacks for Web developers.

Prerequisites

Before you begin:

- Provision a Microsoft Windows Server 2008 R2 or 2012 R2 base instance. You must configure the base instance with a public domain name system (DNS) name that is reachable from the Internet. For more information, see Getting Started with Amazon EC2 Windows Instances (p. 20). Optionally, you might be eligible to configure the base instance on the AWS free tier. The free tier is designed for users with new AWS accounts who want to gain experience with AWS. For more information about the free tier and eligibility requirements, see AWS Free Tier.

  > **Important**
  > If you launch a non-free tier instance and use it to deploy your stack, you will incur the standard Amazon EC2 usage fees for the instance until you terminate it. For more information, see Amazon EC2 Pricing.

- Verify that the security group for your instance has the following ports open:
  - 80 (HTTP inbound and outbound) - Port 80 allows computers outside of the instance to connect by using HTTP.
  - 443 (HTTPS inbound and outbound) - Port 443 allows computers outside of the instance to connect by using HTTPS.
  - 3389 (RDP inbound only) - Port 3389 allows you to connect to the instance with Remote Desktop Protocol (RDP). As a security best practice, restrict RDP access to a range of IP addresses in your organization.

  For more information about these prerequisites, see Setting Up with Amazon EC2 (p. 14).

To install a WAMP server

1. Connect to your instance using Microsoft Remote Desktop. For more information, see Getting Started with Amazon EC2 Windows Instances (p. 20).
2. Disable Internet Explorer Enhanced Security Configuration so that you can download and install required software from the web.

   a. In your Microsoft Windows Server 2008 or 2012 instance, open Server Manager.

      - On Windows Server 2008 R2, under **Server Summary**, in the **Security Information** section, click **Configure IE ESC**.
      - On Windows Server 2012 R2, click **Local Server** in the left pane. In the **Properties** pane, locate **IE Enhanced Security Configuration**. Click **On**.

   b. Under **Administrators**, click **Off**, and then click **OK**.

   c. Close Server Manager.

   > **Note**
   > Make a note to re-enable Internet Explorer Enhanced Security Configuration when you have finished installing software from the web.

3. Install software updates to ensure that the instance has the latest security updates and bug fixes.

a. **EC2Config** - Download and install the latest version of Amazon Windows EC2Config Service.
b. **Windows Update** - Run Windows Update to ensure that the latest security and software updates are installed on the instance. In Control Panel, click **System and Security**. In the **Windows Update** section, click **Check for updates**.

4. Download and install the WAMP stack. For the purposes of this tutorial, we suggest that you download and install this WAMP stack. You can, however, download and install other Bitnami WAMP stacks. Regardless of which stack you install, the Bitnami site prompts you to either create a free Bitnami account or log in by using a social media account. After you log in, run the Bitnami setup wizard.
5. After setup completes, verify that the Apache web server is configured properly and running by browsing to a test page. Open a web browser on a different computer and enter either the public DNS address of the WAMP server or the public IP address. The public DNS address for your instance is listed on the Amazon EC2 console in the **Public DNS** column. If this column is hidden, click the **Show/Hide** icon and select **Public DNS**.

   Important
   If you do not see the Bitnami test page, use Windows Firewall with Advanced Security to create a custom rule that allows the HTTP protocol through port 80 and the HTTPS protocol through port 443. For more information, see Windows Firewall with Advanced Security Overview on Microsoft TechNet. Also verify that the security group you are using contains a rule to allow HTTP (port 80) connections. For information about adding an HTTP rule to your security group, see Adding Rules to a Security Group.
6. Test your WAMP server by viewing a PHP file from the web. You must be logged onto the instance as an administrator to perform the following steps.

   a. Create a file called phpinfo.php containing the code below and place this file in the Apache root directory. By default, the path is: C:\Bitnami\wampstack-*version_number*\apache2\htdocs.

   ```
   <?php phpinfo(); ?>
   ```

   b. In a web browser, enter the URL of the file you just created. This URL is the public DNS address of your instance followed by a forward slash and the file name. For example:

      http://*my.public.dns.amazonaws.com*/phpinfo.php
   c. Verify that the PHP information page is displayed. If the page does not display, verify that you entered the correct public DNS address. Also verify that Windows folder options are configured to show known file extensions. By default, folder options hide known file extensions. If you created the file in Notepad and saved it in the root directory your phpinfo.php file might incorrectly be saved as phpinfo.php.txt.
   d. As a security best practice, delete the phpinfo.php file when you finish testing the WAMP server.

7. Enhance MySQL security by disabling default features and by setting a root password. The **mysql_secure_installation** Perl script can perform these tasks for you. To run the script, you must install Perl.

   a. Download and install Perl from the Perl Programming Language website.
   b. In the **C:\Bitnami\wampstack-version_number\mysql\bin** directory, double-click **mysql_secure_installation**.
   c. When prompted, enter the MySQL root account password that you entered when you ran the Bitnami WAMP stack installer, and then press **Enter**.
   d. Type **n** to skip changing the password.
   e. Type **Y** to remove the anonymous user accounts.
   f. Type **Y** to disable remote root login.

g. Type **Y** to remove the test database.
h. Type **Y** to reload the privilege tables and save your changes.

If you successfully completed the steps in this tutorial, then your WAMP server is functioning properly. To continue testing, you can add more content to the C:\Bitnami\wampstack-*version_number*\apache2\htdocs folder and view the content by using the public DNS address for your instance.

Important
As a best practice, stop the MySQL server if you do not plan to use it right away. You can restart the server when you need it again.

Tutorial: Installing a WIMP Server on an Amazon EC2 Instance Running Windows Server

This tutorial shows you how to install a Microsoft Internet Information Services (IIS) web server with PHP and MySQL on an EC2 instance running Microsoft Windows Server. This software configuration is sometimes called a WIMP server or WIMP stack (Windows, IIS, MySQL, PHP). A WIMP stack is not designed for production environments because MySQL and IIS would compete for server resources. You can, however, create a WIMP stack on an EC2 instance to prototype a web project in a controlled test environment. For example, you can host a static website or deploy a dynamic PHP application that reads and writes information to a database.

Prerequisites

Before you begin:

- Provision a Microsoft Windows Server 2008 R2 or 2012 R2 base instance. You must configure the base instance with a public domain name system (DNS) name that is reachable from the Internet. For more information, see Getting Started with Amazon EC2 Windows Instances (p. 20). Optionally, you might be eligible to configure the base instance using the AWS free tier. The free tier is designed for users with new AWS accounts who want to gain experience with AWS. For more information about the free tier and eligibility requirements, see AWS Free Tier.

  Important
  If you launch a non-free tier instance and use it to deploy your stack, you will incur the standard Amazon EC2 usage fees for the instance until you terminate it. For more information, see Amazon EC2 Pricing.

- Verify that the security group for your instance has the following ports open:
  - 80 (HTTP inbound and outbound) - Port 80 allows computers outside of the instance to connect by using HTTP.
  - 443 (HTTPS inbound and outbound) - Port 443 allows computers outside of the instance to connect by using HTTPS.
  - 3389 (RDP inbound only) - Port 3389 allows you to connect to the instance with Remote Desktop Protocol (RDP). As a security best practice, restrict RDP access to a range of IP addresses in your organization.

  For more information about these prerequisites, see Setting Up with Amazon EC2 (p. 14).
- Read the best practices for installing PHP on the Microsoft web platform.

To install a WIMP server

1. Connect to your instance using Microsoft Remote Desktop. For more information, see Getting Started with Amazon EC2 Windows Instances (p. 20).
2. Disable Internet Explorer Enhanced Security Configuration so that you can download and install required software from the web.

   a. In your Microsoft Windows Server 2008 or 2012 instance, open Server Manager.

      - On Windows Server 2008 R2, under **Server Summary**, in the **Security Information** section, click **Configure IE ESC**.
      - On Windows Server 2012 R2, click **Local Server** in the left pane. In the **Properties** pane, locate **IE Enhanced Security Configuration**. Click **On**.

   b. Under **Administrators**, click **Off**, and then click **OK**.

   c. Close Server Manager.

   Note
   Make a note to re-enable Internet Explorer Enhanced Security Configuration when you have finished installing software from the web.
3. Install software updates to ensure that the instance has the latest security updates and bug fixes.

   a. **EC2Config** - Download and install the latest version of Amazon Windows EC2Config Service.

   b. **Windows Update** - Run Windows Update to ensure that the latest security and software updates are installed on the instance. In Control Panel, click **System and Security**. In the **Windows Update** section, click **Check for updates**.

Install the IIS web server

IIS is a feature of Windows Server and is installed by using Server Manager. This section includes procedures for installing IIS on either Windows Server 2008 or 2012.

Install IIS on Windows Server 2012

1. In **Server Manager** click **Add roles and features**.
2. On the **Before you begin** page, click **Next**.
3. On the **Select installation type** page, select **Role-based or feature-based installation**, and then click **Next**.
4. On the **Select destination server** page, select your instance from the server pool, and then click **Next**.
5. On the **Select server roles** page, select **Web Server (IIS)**, click **Add features**, and then click **Next**.
6. On the **Select features** page, retain the default features and expand **.NET Framework 4.5 Features**, select **ASP.NET 4.5**, and then click **Next**.
7. On the **Web Server Role (IIS)** page, click **Next**.
8. On the **Select role services** page, retain the default services and select **Application Development**.
9. Expand **Application Development**, and then select the following features. When selecting these features, if you are prompted, click **Add features**:

   a. .NET Extensibility 3.5

   b. .NET Extensibility 4.5

   c. Application Initialization

d. ASP.NET 3.5
e. ASP.NET 4.5
f. CGI

10. Click **Next**.
11. On the **Confirm installation selections** page, select **Restart the destination server automatically if required**. When prompted for confirmation, click **Yes**.
12. Click **Install**, and then after the installation is complete, click **Close**.
13. Run Windows update again.

Install IIS on Windows Server 2008

1. In **Server Manager**, click **Roles**.
2. Click **Add Roles**.
3. On the **Before You Begin** page, click **Next**.
4. On the **Select Server Roles** page, click **Web Server (IIS)**.
5. On the **Select Role Services** page under **Application Development**, click **ASP.NET**.

   a. When prompted, click **Add Required Role Services**.
   b. Click **CGI**.
   c. Click **Next**.

6. On the **Confirm Installation Selections**, click **Install**.
7. Run Windows update again.

Verify that the web server is running

After setup completes, verify that the IIS web server is configured properly and running by going to the IIS welcome page. Open a web browser on a different computer and enter either the public DNS address of the WIMP server or the public IP address. The public DNS address for your instance is listed on the Amazon EC2 console in the **Public DNS** column. If this column is hidden, click the **Show/Hide** icon and select **Public DNS**.

> **Important**
> If you do not see the IIS welcome page, use Windows Firewall with Advanced Security to create a custom rule that allows the HTTP protocol through port 80 and the HTTPS protocol through port 443. For more information, see Windows Firewall with Advanced Security Overview on Microsoft TechNet. Also verify that the security group you are using contains a rule to allow HTTP (port 80) connections. For information about adding an HTTP rule to your security group, see Adding Rules to a Security Group.

Install MySQL and PHP

You can download and install MySQL and PHP by using the Microsoft Web Platform Installer, as described in this section.

To install MySQL and PHP

1. In your Microsoft Windows Server instance, download and install the latest version of the Microsoft Web Platform Installer 5.0.
2. In the Microsoft Web Platform Installer click the **Products** tab.
3. Select **MySQL Windows 5.5** and click **Add**.

4. Select **PHP 5.6.0** and click **Add**.
5. Click **Install**.
6. On the **Prerequisites** page, enter a password for the MySQL default database administrator account, and then click **Continue**.
7. When the installation is complete, click **Finish**, and then click **Exit** to close the Web Platform Installer.

Test your WIMP server

Test your WIMP server by viewing a PHP file from the web. You must be logged onto the instance as an administrator to perform the following steps.

To test your WIMP server

1. Download and install the Visual C++ Redistributable for Visual Studio 2012 Update 4 x86 package. Even if your server is a 64-bit server, you must install the x86 package.
2. Create a file called phpinfo.php that contains the following code and place this file in the IIS root directory. By default, the path is: C:\inetpub\wwwroot.

```
<?php phpinfo(); ?>
```

3. In a web browser, enter the URL of the file you just created. This URL is the public DNS address of your instance followed by a forward slash and the file name, as in the following example:

   http://*my.public.dns.amazonaws.com*/phpinfo.php
4. Verify that the PHP information page is displayed. If the page does not display, verify that you entered the correct public DNS address. Also verify that Windows folder options are configured to show known file extensions. By default, folder options hide known file extensions. If you created the file in Notepad and saved it in the root directory your phpinfo.php file might incorrectly be saved as phpinfo.php.txt.
5. As a security best practice, delete the phpinfo.php file when you finish testing the WAMP server.
6. Enhance MySQL security by disabling default features and by setting a root password. The **mysql_secure_installation** Perl script can perform these tasks for you. To run the script, you must install Perl.

   a. Download and install Perl from the Perl Programming Language website.
   b. In the C:\Program Files\MySQL\MySQL Server 5.5\bin directory, double-click **mysql_secure_installation**.
   c. When prompted, enter the current root password and press **Enter**.
   d. Type **n** to skip changing the password.
   e. Type **Y** to remove the anonymous user accounts.
   f. Type **Y** to disable remote root login.
   g. Type **Y** to remove the test database.
   h. Type **Y** to reload the privilege tables and save your changes.

   You should now have a fully functional WIMP web server. If you add content to the IIS document root at C:\inetpub\wwwroot, you can view that content at the public DNS address for your instance.

   Important
   As a best practice, stop the MySQL server if you do not plan to use it right away. You can restart the server when you need it again.

Tutorial: Setting Up a Windows HPC Cluster on Amazon EC2

You can launch a scalable Microsoft Windows High Performance Computing (HPC) cluster using EC2 instances. A Windows HPC cluster requires an Active Directory domain controller, a DNS server, a head node, and one or more compute nodes.

To set up a Windows HPC cluster on Amazon EC2, complete the following tasks:

- Task 1: Set Up Your Active Directory Domain Controller (p. 41)
- Task 2: Configure Your Head Node (p. 43)
- Task 3: Set Up the Compute Node (p. 45)
- Task 4: Scale Your HPC Compute Nodes (Optional) (p. 46)

For more information about high performance computing, see High Performance Computing (HPC) on AWS.

Prerequisites

Install the Amazon EC2 command line interface tools and set the region you'll be using as the default region. For more information, see Setting Up the Amazon EC2 Command Line Interface Tools on Windows in the *Amazon EC2 Command Line Reference*.

Task 1: Set Up Your Active Directory Domain Controller

The Active Directory domain controller provides authentication and centralized resource management of the HPC environment and is required for the installation. To set up your Active Directory, complete these steps:

1. Create the security groups required for Active Directory.
2. Create the instance that serves as the domain controller for your HPC cluster.
3. Configure the domain controller for your HPC cluster.

Creating Security Groups for Active Directory

Run the script `Create-AD-sec-groups.bat` to create a security group with rules for the domain controller and domain members.

To create the required security groups for Active Directory

1. Copy the contents of Create_AD_security.bat (p. 47) to a text editor. Save the file, using the file name `Create-AD-sec-groups.bat`, to a computer configured with the Amazon EC2 command line interface tools.
2. Run the `Create-AD-sec-groups.bat` batch file from the Command Prompt window as a local administrator.
3. Open the Amazon EC2 console, select **Security Groups** from the navigation pane, and verify that the following security groups appear in the list:

- `SG - Domain Controller`
- `SG - Domain Member`

Alternatively, manually set up the firewall to allow traffic on the required ports. For more information, see How to configure a firewall for domains and trusts on the Microsoft website.

Creating the Domain Controller for your HPC cluster

Launch an instance that will serve as the domain controller for your HPC cluster.

To create a domain controller for your HPC cluster

1. Open the Amazon EC2 console and select a region for the instance.
2. Launch an instance with the name **`Domain Controller`** and the security group **`SG - Domain Controller`**.
    a. On the console dashboard, click **Launch Instance**.
    b. On the **Choose an AMI** page, select an AMI for Windows Server and then click **Select**.
    c. On the next pages of the wizard, select an instance type, instance configuration, and storage options.
    d. On the **Tag Instance** page, enter `Domain Controller` as the value for the `Name` tag and then click **Next: Configure Security Group**.
    e. On the **Configure Security Group** page, click **Select an existing security group**, select `SG - Domain Controller` from the list of security groups, and then click **Review and Launch**.
    f. Click **Launch**.
3. Create an Elastic IP address and associate it with the instance.
    a. In the navigation pane, click **Elastic IPs**.
    b. Click **Allocate New Address**.
    c. When prompted, click **Yes, Allocate**, and then close the confirmation dialog box.
    d. Select the Elastic IP address you created, and then click **Associate Address**.
    e. In the **Instance** list, select the `Domain Controller` instance and then click **Associate**.

Configuring the Domain Controller for Your HPC Cluster

Log in to the instance you created and configure the server as a domain controller for the HPC cluster.

To configure your instance as a domain controller

1. Connect to your `Domain Controller` instance.
2. Open **Server Manager**, and add the Active Directory Domain Services role.
3. Promote the server to a domain controller using Server Manager or by running **DCPromo.exe**.
4. Create a new domain in a new forest.
5. Enter **`hpc.local`** as the fully qualified domain name (FQDN).
6. Select Forest Functional Level as **Windows Server 2008 R2**.
7. Ensure that the DNS Server option is selected, and then click **Next**.

8. Select **Yes, the computer will use an IP address automatically assigned by a DHCP server (not recommended)**.
9. In the warning box, click **Yes** to continue.
10. Complete the wizard and then select **Reboot on Completion**.
11. Log in to the instance as **`hpc.local\administrator`**.
12. Create a domain user **`hpc.local\hpcuser`**.

Task 2: Configure Your Head Node

An HPC client connects to the head node. The head node facilitates the scheduled jobs. You configure your head node by completing the following steps:

1. Create security groups for your HPC cluster.
2. Launch an instance for your head node.
3. Install the HPC Pack.
4. Configure your HPC cluster.

Creating Security Groups for Your HPC Cluster

Run the script `Create-HPC-sec-group.bat` to create a security group named `SG - Windows HPC Cluster` with rules for the HPC cluster nodes.

To create the security group for your HPC cluster

1. Copy the contents of Create-HPC-sec-group.bat (p. 48) to a text editor. Save the file, using the file name `Create-HPC-sec-group.bat`, to a computer configured with the EC2 command line tools.
2. Run the `Create-HPC-sec-group.bat` batch file from a Command Prompt window as a local administrator.
3. Open the Amazon EC2 console, select **Security Groups** from the navigation pane, and verify that the `SG - Windows HPC Cluster` security group appears in the list.

Alternatively, manually configure the firewall with the port requirements for HPC cluster members to communicate. For more information, see Windows Firewall configuration on the Microsoft website.

Launch an Instance for the HPC Head Node

Launch an instance and then configure it as a member of the `hpc.local` domain and with the necessary user accounts.

To configure an instance as your head node

1. Launch an instance and name it **`HPC-Head`**. When you launch the instance, select both of these security groups:
   - `SG - Windows HPC Cluster`
   - `SG - Domain Member`
2. Log in to the instance and get the existing DNS server address from **HPC-Head** using the following command:

```
C:\> IPConfig /all
```

3. Update the TCP/IPv4 properties of the `HPC-Head` NIC to include the Elastic IP address for the `Domain Controller` instance as the primary DNS, and then add the additional DNS IP address from the previous step.
4. Join the machine to the `hpc.local` domain using the credentials for `hpc.local\administrator` (the domain administrator account).
5. Add `hpc.local\hpcuser` as the local administrator. When prompted for credentials, use `hpc.local\administrator`, and then restart the instance.
6. Log back in to `HPC-Head` as `hpc.local\hpcuser`.

Install the HPC Pack

To install the HPC Pack

1. Connect to your **HPC-Head** instance using the `hpc.local\hpcuser` account.
2. Using **Server Manager**, turn off Internet Explorer Enhanced Security Configuration (IE ESC) for Administrators.
   a. In **Server Manager**, under **Security Information**, click **Configure IE ESC**.
   b. Turn off IE ESC for administrators.
3. Install the HPC Pack on **HPC-Head**.
   a. Download the HPC Pack to `HPC-Head` from the Microsoft Download Center. Choose the HPC Pack for the version of Windows Server on `HPC-Head`.
   b. Extract the files to a folder, open the folder, and double-click **setup.exe**.
   c. On the Installation page, select **Create a new HPC cluster by creating a head node**, and then click **Next**.
   d. Accept the default settings to install all the databases on the Head Node, and then click **Next**.
   e. Complete the wizard.

Configure Your HPC Cluster on the Head Node

To configure your HPC cluster on the head node

1. Start **HPC Cluster Manager**.
2. In the **Deployment To-Do List**, select **Configure your network**.
   a. In the wizard, select the default option (5), and then click **Next**.
   b. Complete the wizard accepting default values on all screens, and choose how you want to update the server and participate in customer feedback.
   c. Click **Configure**.
3. Select **Provide Network Credentials**, then supply the `hpc.local\hpcuser` credentials.
4. Select **Configure the naming of new nodes**, and then click **OK**.
5. Select **Create a node template**.

a. Select the **Compute node template**, and then click **Next**.
b. Select **Without operating system**, and then continue with the defaults.
c. Click **Create**.

Task 3: Set Up the Compute Node

Setting up the compute node involves the following steps:

1. Launch an instance for your compute node.
2. Install the HPC Pack on the instance.
3. Add the compute node to your cluster.

Launch an Instance for the HPC Compute Node

Configure your compute node by launching an instance, and then configuring the instance as a member of the `hpc.local` domain with the necessary user accounts.

To configure an instance for your compute node

1. Launch an instance and name it **`HPC-Compute`**. When you launch the instance, select the following security groups: **SG - Windows HPC Cluster** and **SG - Domain Member**.
2. Log in to the instance and get the existing DNS server address from **HPC-Compute** using the following command:

```
C:\> IPConfig /all
```

3. Update the TCP/IPv4 properties of the `HPC-Compute` NIC to include the Elastic IP address of the `Domain Controller` instance as the primary DNS. Then add the additional DNS IP address from the previous step.
4. Join the machine to the `hpc.local` domain using the credentials for `hpc.local\administrator` (the domain administrator account).
5. Add `hpc.local\hpcuser` as the local administrator. When prompted for credentials, use `hpc.local\administrator`, and then restart.
6. Log back in to `HPC-Compute` as `hpc.local\hpcuser`.

Install the HPC Pack on the Compute Node

To install the HPC Pack on the compute node

1. Connect to your `HPC-Compute` instance using the `hpc.local\hpcuser` account.
2. Using **Server Manager**, turn off Internet Explorer Enhanced Security Configuration (IE ESC) for Administrators.
   a. In **Server Manager**, under **Security Information**, click **Configure IE ESC**.
   b. Turn off IE ESC for administrators.
3. Install the HPC Pack on `HPC-Compute`.

a. Download the HPC Pack to `HPC-Compute` from the Microsoft Download Center. Choose the HPC Pack for the version of Windows Server on `HPC-Compute`.
b. Extract the files to a folder, open the folder, and double-click **setup.exe**.
c. On the **Installation** page, select **Join an existing HPC cluster by creating a new compute node**, and then click **Next**.
d. Specify the fully-qualified name of the `HPC-Head` instance, and then choose the defaults.
e. Complete the wizard.

Add the Compute Node to Your HPC Cluster

To complete your cluster configuration, from the head node, add the compute node to your cluster.

To add the compute node to your cluster

1. Connect to the `HPC-Head` instance as `hpc.local\hpcuser`.
2. Open **HPC Cluster Manager**.
3. Select **Node Management**.
4. If the compute node displays in the **Unapproved** bucket, right-click the node that is listed and select **Add Node**.
   a. Select **Add compute nodes or broker nodes that have already been configured**.
   b. Select the check box next to the node and click **Add**.
5. Right-click the node and click **Bring Online**.

Task 4: Scale Your HPC Compute Nodes (Optional)

To scale your compute nodes

1. Connect to the `HPC-Compute` instance as `hpc.local\hpcuser`.
2. Delete any files you downloaded locally from the HP Pack installation package. (You have already run setup and created these files on your image so they do not need to be cloned for an AMI.)
3. From `C:\Program Files\Amazon\Ec2ConfigService` open the file `sysprep2008.xml`.
4. At the bottom of `<settings pass="specialize">`, add the following section. Make sure to replace *hpc.local*, *password*, and *hpcuser* to match your environment.

```
<component name="Microsoft-Windows-UnattendedJoin" processorArchitec
ture="amd64" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.mi
crosoft.com/WMIConfig/2002/State"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Identification>
        <UnsecureJoin>false</UnsecureJoin>
        <Credentials>
            <Domain>hpc.local</Domain>
            <Password>password</Password>
            <Username>hpcuser</Username>
        </Credentials>
        <JoinDomain>hpc.local</JoinDomain>
```

```
    </Identification>
</component>
```

5. Save `sysprep2008.xml`.
6. Click **Start**, point to **All Programs**, and then click **EC2ConfigService Settings**.
   a. Click the **General** tab, and clear the **Set Computer Name** check box.
   b. Click the **Bundle** tab, and then click **Run Sysprep and Shutdown Now**.
7. Open the Amazon EC2 console.
8. In the navigation pane, click **Instances**.
9. Wait for the instance status to show **stopped**.
10. Right-click the instance, select **Image**, and select **Create Image**.
11. Specify an image name and image description, and then click **Create Image** to create an AMI from the instance.
12. Start the original `HPC-Compute` instance that was shut down.
13. Connect to the head node using the `hpc.local\hpcuser` account.
14. From **HPC Cluster Manager**, delete the old node that now appears in an error state.
15. In the Amazon EC2 console, in the navigation pane, click **AMIs**.
16. Use the AMI you created to add additional nodes to the cluster.

You can launch additional compute nodes from the AMI that you created. These nodes are automatically joined to the domain, but you must add them to the cluster as already configured nodes in **HPC Cluster Manager** using the head node and then bring them online.

Running the Lizard Performance Measurement Application

If you choose, you can run the Lizard application, which measures the computational performance and efficiency that can be achieved by your HPC cluster. Go to http://www.microsoft.com/download/en/details.aspx?id=8433, download the lizard_x64.msi installer, and run the installer directly on your head node as `hpc.local\hpcuser`.

Create_AD_security.bat

The following batch file creates two security groups for your Active Directory environment: one group for Active Directory domain controllers and one for Active Directory domain member servers.

```
set DC="SG - Domain Controller"
set DM="SG - Domain Member"
set CIDR="your-address-range"

:: ========================================================================
:: Creates Security groups Prior to Adding Rules
:: ========================================================================

call ec2addgrp %DM% -d "Active Directory Domain Member"
call ec2addgrp %DC% -d "Active Directory Domain Controller"
```

```
:: ================================================================================
:: Security group for Domain Controller
:: ================================================================================

:: For LDAP and related services. Details at link below
:: http://support.microsoft.com/kb/179442
call ec2auth %DC% -o %DM% -P UDP -p 123
call ec2auth %DC% -o %DM% -P TCP -p 135
call ec2auth %DC% -o %DM% -P UDP -p 138
call ec2auth %DC% -o %DM% -P TCP -p "49152-65535"
call ec2auth %DC% -o %DM% -P TCP -p 389
call ec2auth %DC% -o %DM% -P UDP -p 389
call ec2auth %DC% -o %DM% -P TCP -p 636
call ec2auth %DC% -o %DM% -P TCP -p 3268
call ec2auth %DC% -o %DM% -P TCP -p 3269
call ec2auth %DC% -o %DM% -P TCP -p 53
call ec2auth %DC% -o %DM% -P UDP -p 53
call ec2auth %DC% -o %DM% -P TCP -p 88
call ec2auth %DC% -o %DM% -P UDP -p 88
call ec2auth %DC% -o %DM% -P TCP -p 445
call ec2auth %DC% -o %DM% -P UDP -p 445

:: For ICMP as required by Active Directory
call ec2auth %DC% -P ICMP -t -1:-1

:: For Elastic IP to communicate with DNS
call ec2auth %DC% -s %CIDR% -P UDP -p 53

:: For RDP for connecting to desktop remotely
call ec2auth %DC% -s %CIDR% -P TCP -p 3389

:: ================================================================================
:: Security group for Domain Member
:: ================================================================================

:: For LDAP and related services. Details at link below
:: http://support.microsoft.com/kb/179442

call ec2auth %DM% -o %DC% -P TCP -p "49152-65535"
call ec2auth %DM% -o %DC% -P UDP -p "49152-65535"
call ec2auth %DM% -o %DC% -P TCP -p 53
call ec2auth %DM% -o %DC% -P UDP -p 53
```

Create-HPC-sec-group.bat

The following batch file creates a security group for your HPC cluster nodes.

```
set HPC="SG - Windows HPC Cluster"
set CIDR="your-address-range"

:: ================================================================================
:: Creates Security groups Prior to Adding Rules
:: ================================================================================

call ec2addgrp %HPC% -d "Windows HPC Server 2008 R2 Cluster Nodes"
```

```
:: ============================================================================
:: Security group for Windows HPC Cluster
:: ============================================================================

:: For HPC related services. Details at link below
:: http://technet.microsoft.com/en-us/library/ff919486.aspx#BKMK_Firewall
call ec2auth %HPC% -o %HPC% -P TCP -p 80
call ec2auth %HPC% -o %HPC% -P TCP -p 443
call ec2auth %HPC% -o %HPC% -P TCP -p 1856
call ec2auth %HPC% -o %HPC% -P TCP -p 5800
call ec2auth %HPC% -o %HPC% -P TCP -p 5801
call ec2auth %HPC% -o %HPC% -P TCP -p 5969
call ec2auth %HPC% -o %HPC% -P TCP -p 5970
call ec2auth %HPC% -o %HPC% -P TCP -p 5974
call ec2auth %HPC% -o %HPC% -P TCP -p 5999
call ec2auth %HPC% -o %HPC% -P TCP -p 6729
call ec2auth %HPC% -o %HPC% -P TCP -p 6730
call ec2auth %HPC% -o %HPC% -P TCP -p 7997
call ec2auth %HPC% -o %HPC% -P TCP -p 8677
call ec2auth %HPC% -o %HPC% -P TCP -p 9087
call ec2auth %HPC% -o %HPC% -P TCP -p 9090
call ec2auth %HPC% -o %HPC% -P TCP -p 9091
call ec2auth %HPC% -o %HPC% -P TCP -p 9092
call ec2auth %HPC% -o %HPC% -P TCP -p "9100-9163"
call ec2auth %HPC% -o %HPC% -P TCP -p "9200-9263"
call ec2auth %HPC% -o %HPC% -P TCP -p 9794
call ec2auth %HPC% -o %HPC% -P TCP -p 9892
call ec2auth %HPC% -o %HPC% -P TCP -p 9893
call ec2auth %HPC% -o %HPC% -P UDP -p 9893

:: For HPC related services, these are NOT in the first table but are there in
 the third table at link below
:: http://technet.microsoft.com/en-us/library/ff919486.aspx#BKMK_Firewall
call ec2auth %HPC% -o %HPC% -P TCP -p 6498
call ec2auth %HPC% -o %HPC% -P TCP -p 7998
call ec2auth %HPC% -o %HPC% -P TCP -p 8050
call ec2auth %HPC% -o %HPC% -P TCP -p 5051

:: For RDP for connecting to desktop remotely
call ec2auth %HPC% -s %CIDR% -P TCP -p 3389
```

Amazon Machine Images (AMI)

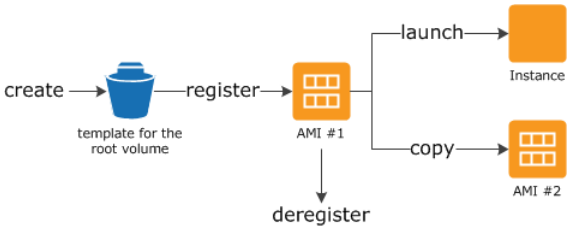
An Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud. You specify an AMI when you launch an instance, and you can launch as many instances from the AMI as you need. You can also launch instances from as many different AMIs as you need.

An AMI includes the following:

- A template for the root volume for the instance (for example, an operating system, an application server, and applications)
- Launch permissions that control which AWS accounts can use the AMI to launch instances
- A block device mapping that specifies the volumes to attach to the instance when it's launched

Using an AMI

The following diagram summarizes the AMI lifecycle. After you create and register an AMI, you can use it to launch new instances. (You can also launch instances from an AMI if the AMI owner grants you launch permissions.) You can copy an AMI to the same region or to different regions. When you are finished launching instance from an AMI, you can deregister the AMI.



You can search for an AMI that meets the criteria for your instance. You can search for AMIs provided by AWS or AMIs provided by the community. For more information, see AMI Types (p. 52) and Finding a Windows AMI (p. 56).

When you are connected to an instance, you can use it just like you use any other server. For information about launching, connecting, and using your instance, see Amazon EC2 Instances (p. 96).

Creating Your Own AMI

You can customize the instance that you launch from a public AMI and then save that configuration as a custom AMI for your own use. Instances that you launch from your AMI use all the customizations that you've made.

The root storage device of the instance determines the process you follow to create an AMI. The root volume of an instance is either an Amazon EBS volume or an instance store volume. For information, see Root Device Volume (p. 8).

To create an Amazon EBS-backed AMI, see Creating an Amazon EBS-Backed Windows AMI (p. 68). To create an instance store-backed AMI, see Creating an Instance Store-Backed Windows AMI (p. 70).

To help categorize and manage your AMIs, you can assign custom *tags* to them. For more information, see Tagging Your Amazon EC2 Resources (p. 609).

Buying, Sharing, and Selling AMIs

After you create an AMI, you can keep it private so that only you can use it, or you can share it with a specified list of AWS accounts. You can also make your custom AMI public so that the community can use it. Building a safe, secure, usable AMI for public consumption is a fairly straightforward process, if you follow a few simple guidelines. For information about how to create and use shared AMIs, see Shared AMIs (p. 58).

You can purchase an AMIs from a third party, including AMIs that come with service contracts from organizations such as Red Hat. You can also create an AMI and sell it to other Amazon EC2 users. For more information about buying or selling AMIs, see Paid AMIs (p. 64).

Deregistering Your AMI

You can deregister an AMI when you have finished with it. After you deregister an AMI, you can't use it to launch new instances. For more information, see Deregistering Your AMI (p. 76).

AWS Windows AMIs

AWS provides a set of publicly available AMIs that contain software configurations specific to the Windows platform. Using these AMIs, you can quickly start building and deploying your applications using Amazon EC2. First choose the AMI that meets your specific requirements, and then launch an instance using that AMI. You retrieve the password for the administrator account and then log in to the instance using Remote Desktop Connection, just as you would with any other Windows server. The name of the administrator account depends on the language of the operating system. For example, for English, it's Administrator, for French it's Administrateur, and for Portuguese it's Administrador. For more information, see Localized Names for Administrator Account in Windows in the Microsoft TechNet Wiki.

Selecting an Initial Windows AMI

To view the Windows AMIs provided by AWS using the Amazon EC2 console, click this link to filter the list of public AMIs: Windows AMIs. If you launch an instance using the Amazon EC2 console, the first page of the wizard includes a **Quick Start** tab that lists some of the most popular AMIs provided by AWS, including AMIs that are eligible for the free tier.

AWS currently provides AMIs based on the following versions of Windows:

- Microsoft Windows Server 2012 R2 (64-bit)
- Microsoft Windows Server 2012 (64-bit)
- Microsoft Windows Server 2008 R2 (64-bit)
- Microsoft Windows Server 2008 (64-bit)
- Microsoft Windows Server 2008 (32-bit)
- Microsoft Windows Server 2003 R2 (64-bit)
- Microsoft Windows Server 2003 R2 (32-bit)

Some of these AMIs also include an edition of Microsoft SQL Server (SQL Enterprise Edition, SQL Server Standard, SQL Server Express, or SQL Server Web). Launching an instance from an AWS Windows AMI with Microsoft SQL Server enables you to run the instance as a database server. Alternatively, you can launch an instance from any Windows AMI and then install the database software that you need on the instance. To view Windows Server AMIs with SQL Server, see Windows AMIs on the AWS Marketplace.

Some AMIs come with Internet Information Services (IIS) and ASP.NET already configured, to help you get started quickly. Alternatively, you can launch an instance from any Windows AMI and then install IIS and ASP.NET. For step-by-step directions, see Configure Your EC2 Instance in *Getting Started with AWS: Hosting a .NET Web App*.

In addition to the public AMIs provided by AWS, AMIs published by the AWS developer community are available for your use. We highly recommend that you use only those Windows AMIs that AWS or other reputable sources provide. To learn how to find a list of Microsoft Windows AMIs approved by Amazon, see Finding a Windows AMI (p. 56).

You can also create an AMI from your own Windows computer. For more information, see Importing and Exporting Instances (p. 169).

Keeping Your AMIs Up-to-Date

AWS provides updated, fully-patched Windows AMIs within five business days of Microsoft's patch Tuesday (the second Tuesday of each month). For more information, see AWS Windows AMI Versions (p. 80).

At their initial launch, your Windows instances contain all the latest security updates. We recommend that you run the Windows Update service as a first step after you launch a Windows, and before you create an AMI. After you launch an instance or create an AMI, you are responsible for keeping them up-to-date. You can use the Windows Update service, or the Automatic Updates tool available on your instance to deploy Microsoft updates to your instance. You must also keep any other software that you deploy to your instance up-to-date using whatever mechanism is appropriate for that software. After you update your Windows instance, you can create an AMI that replaces any previous AMIs that you created. For more information, see Updating Your Windows Instance (p. 79).

AMI Types

You can select an AMI to use based on the following characteristics:

- Region (see Regions and Availability Zones (p. 6))
- Operating system
- Architecture (32-bit or 64-bit)

- Launch Permissions (p. 53)
- Storage for the Root Device (p. 53)

Launch Permissions

The owner of an AMI determines its availability by specifying launch permissions. Launch permissions fall into the following categories.

Launch Permission	Description
public	The owner grants launch permissions to all AWS accounts.
explicit	The owner grants launch permissions to specific AWS accounts.
implicit	The owner has implicit launch permissions for an AMI.

Amazon and the Amazon EC2 community provide a large selection of public AMIs. For more information, see Shared AMIs (p. 58). Developers can charge for their AMIs. For more information, see Paid AMIs (p. 64).

Storage for the Root Device

All AMIs are categorized as either *backed by Amazon EBS* or *backed by instance store*. The former means that the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot. The latter means that the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3. For more information, see Root Device Volume (p. 8).

This section summarizes the important differences between the two types of AMIs. The following table provides a quick summary of these differences.

Characteristic	Amazon EBS-Backed	Amazon Instance Store-Backed
Boot time	Usually less than 1 minute	Usually less than 5 minutes
Size limit	16 TiB	10 GiB
Root device volume	Amazon EBS volume	Instance store volume
Data persistence	By default, the root volume is deleted when the instance terminates.* Data on any other Amazon EBS volumes persists after instance termination by default. Data on any instance store volumes persists only during the life of the instance.	Data on any instance store volumes persists only during the life of the instance. Data on any Amazon EBS volumes persists after instance termination by default.
Upgrading	The instance type, kernel, RAM disk, and user data can be changed while the instance is stopped.	Instance attributes are fixed for the life of an instance.

Characteristic	Amazon EBS-Backed	Amazon Instance Store-Backed
Charges	You're charged for instance usage, Amazon EBS volume usage, and storing your AMI as an Amazon EBS snapshot.	You're charged for instance usage and storing your AMI in Amazon S3.
AMI creation/bundling	Uses a single command/call	Requires installation and use of AMI tools
Stopped state	Can be placed in stopped state where instance is not running, but the root volume is persisted in Amazon EBS	Cannot be in stopped state; instances are running or terminated

* By default, Amazon EBS-backed instance root volumes have the `DeleteOnTermination` flag set to `true`. For information about how to change this flag so that the volume persists after termination, see Root Device Volume (p. 8).

Determining the Root Device Type of Your AMI

To determine the root device type of an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**, and select the AMI.
3. Check the value of **Root Device Type** in the **Details** tab as follows:
   - If the value is `ebs`, this is an Amazon EBS-backed AMI.
   - If the value is `instance store`, this is an instance store-backed AMI.

To determine the root device type of an AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-images (AWS CLI)
- ec2-describe-images (Amazon EC2 CLI)
- Get-EC2Image (AWS Tools for Windows PowerShell)

Size Limit

Amazon EC2 instance store-backed AMIs are limited to 10 GiB storage for the root device, whereas Amazon EBS-backed AMIs are limited to 1 TiB. Many Windows AMIs come close to the 10 GiB limit, so you'll find that Windows AMIs are often backed by an Amazon EBS volume.

Note
All Windows Server 2008, Windows Server 2008 R2, and Windows Server 2012 AMIs are backed by an Amazon EBS volume by default because of their larger size.

Stopped State

You can stop an Amazon EBS-backed instance, but not an Amazon EC2 instance store-backed instance. Stopping causes the instance to stop running (its status goes from `running` to `stopping` to `stopped`).

A stopped instance persists in Amazon EBS, which allows it to be restarted. Stopping is different from terminating; you can't restart a terminated instance. Because Amazon EC2 instance store-backed AMIs can't be stopped, they're either running or terminated. For more information about what happens and what you can do while an instance is stopped, see Stop and Start Your Instance (p. 218).

Default Data Storage and Persistence

Instances that use an instance store volume for the root device automatically have instance store available (the root volume contains the root partition and you can store additional data). Any data on an instance store volume is deleted when the instance fails or terminates (except for data on the root device). You can add persistent storage to your instance by attaching one or more Amazon EBS volumes.

Instances that use Amazon EBS for the root device automatically have an Amazon EBS volume attached. The volume appears in your list of volumes like any other. The instances don't use any available instance store volumes by default. You can add instance storage or additional Amazon EBS volumes using a block device mapping. For more information, see Block Device Mapping (p. 587). For information about what happens to the instance store volumes when you stop an instance, see Stop and Start Your Instance (p. 218).

Boot Times

Amazon EBS-backed AMIs launch faster than Amazon EC2 instance store-backed AMIs. When you launch an Amazon EC2 instance store-backed AMI, all the parts have to be retrieved from Amazon S3 before the instance is available. With an Amazon EBS-backed AMI, only the parts required to boot the instance need to be retrieved from the snapshot before the instance is available. However, the performance of an instance that uses an Amazon EBS volume for its root device is slower for a short time while the remaining parts are retrieved from the snapshot and loaded into the volume. When you stop and restart the instance, it launches quickly, because the state is stored in an Amazon EBS volume.

AMI Creation

To create Windows AMIs backed by instance store, there's an API action that creates an AMI and another API action that registers the AMI.

AMI creation is much easier for AMIs backed by Amazon EBS. The `CreateImage` API action creates your Amazon EBS-backed AMI and registers it. There's also a button in the AWS Management Console that lets you create an AMI from a running instance. For more information, see Creating an Amazon EBS-Backed Windows AMI (p. 68).

How You're Charged

With AMIs backed by instance store, you're charged for AMI storage and instance usage. With AMIs backed by Amazon EBS, you're charged for volume storage and usage in addition to the AMI and instance usage charges.

With Amazon EC2 instance store-backed AMIs, each time you customize an AMI and create a new one, all of the parts are stored in Amazon S3 for each AMI. So, the storage footprint for each customized AMI is the full size of the AMI. For Amazon EBS-backed AMIs, each time you customize an AMI and create a new one, only the changes are stored. So the storage footprint for subsequent AMIs you customize after the first is much smaller, resulting in lower AMI storage charges.

When an Amazon EBS-backed instance is stopped, you're not charged for instance usage; however, you're still charged for volume storage. We charge a full instance hour for every transition from a stopped state to a running state, even if you transition the instance multiple times within a single hour. For example, let's say the hourly instance charge for your instance is $0.10. If you were to run that instance for one hour without stopping it, you would be charged $0.10. If you stopped and restarted that instance twice

during that hour, you would be charged $0.30 for that hour of usage (the initial $0.10, plus 2 x $0.10 for each restart).

Finding a Windows AMI

Before you can launch an instance, you must select an AMI to use. As you select an AMI, consider the following requirements you might have for the instances that you'll launch:

- The region
- The operating system (see AWS Windows AMIs (p. 51))
- The architecture: 32-bit (`i386`) or 64-bit (`x86_64`)
- The root device type: Amazon EBS or instance store
- The provider: Amazon Web Services, Oracle, IBM, Microsoft, or the community

If you need to find a Linux AMI, see Finding a Linux AMI in the *Amazon EC2 User Guide for Linux Instances*.

Contents

- Finding a Windows AMI Using the Amazon EC2 Console (p. 56)
- Finding an AMI Using the AWS CLI (p. 57)
- Finding an AMI Using the Amazon EC2 CLI (p. 57)
- Finding an AMI Using the AWS Tools for Windows PowerShell (p. 57)
- Finding a Windows Server 2003 AMI (p. 58)

Finding a Windows AMI Using the Amazon EC2 Console

You can find Windows AMIs using the Amazon EC2 console. You can search through all available AMIs using the **Images** page, or select from commonly used AMIs on the **Quick Launch** tab when you use the console to launch an instance.

To find a Windows AMI using the Images page

1. Open the Amazon EC2 console.
2. From the navigation bar, select a region. You can select any region that's available to you, regardless of your location. This is the region in which you'll launch your instance.
3. In the navigation pane, click **AMIs**.
4. (Optional) Use the **Filter** options to scope the list of displayed AMIs to see only the AMIs that interest you. For example, to list all Windows AMIs provided by AWS, select **Public images**. Click the Search bar and select **Owner** from the menu, then select **Amazon images**. Click the Search bar again to select **Platform** and then the operating system from the list provided.
5. (Optional) Click the **Show/Hide Columns** icon to select which image attributes to display, such as the root device type. Alternatively, you can select an AMI from the list and view its properties in the **Details** tab.
6. To launch an instance from this AMI, select it and then click **Launch**. For more information about launching an instance using the console, see Launching Your Instance from an AMI (p. 208). If you're not ready to launch the instance now, write down the AMI ID (ami-*xxxxxxxx*) for later.

To find a Windows AMI when you launch an instance

1. Open the Amazon EC2 console.
2. From the console dashboard, click **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, on the **Quick Start** tab, select from one of the commonly used AMIs in the list. If you don't see the AMI that you need, select the **AWS Marketplace** or **Community AMIs** tab to find additional AMIs.

Finding an AMI Using the AWS CLI

You can use command line parameters to list only the types of AMIs that interest you. For example, you can use the describe-images command as follows to find public AMIs owned by you or Amazon.

```
C:\> aws ec2 describe-images --owners self amazon
```

Add the following filter to the previous command to display only Windows AMIs:

```
--filters "Name=platform,Values=windows"
```

After locating an AMI that meets your needs, write down its ID (ami-*xxxxxxxx*). You can use this AMI to launch instances. For more information, see Launching an Instance Using the AWS CLI in the *AWS Command Line Interface User Guide*.

Finding an AMI Using the Amazon EC2 CLI

You can use command line parameters to list only the types of AMIs that interest you. For example, you can use the ec2-describe-images command as follows to find public AMIs owned by you or Amazon.

```
C:\> ec2-describe-images -o self -o amazon
```

Add the following filter to the previous command to display only Windows AMIs:

```
--filter "platform=windows"
```

After locating an AMI that meets your needs, write down its ID (ami-*xxxxxxxx*). You can use this AMI to launch instances. For more information, see Launching an Instance Using the Amazon EC2 CLI in the *Amazon EC2 Command Line Reference*.

Finding an AMI Using the AWS Tools for Windows PowerShell

You can use command-line parameters to list only the types of AMIs that interest you. For more information, see Find an AMI Using Windows PowerShell in the *AWS Tools for Windows PowerShell User Guide*.

After locating an AMI that meets your needs, write down its ID (ami-*xxxxxxxx*). You can use this AMI to launch instances. For more information, see Launch an Instance Using Windows PowerShell in the *AWS Tools for Windows PowerShell User Guide*.

Finding a Windows Server 2003 AMI

Beginning July 14, 2015, Microsoft will no longer support Windows Server 2003. If your business or organization is currently running Windows Server 2003 EC2 instances, we recommend that you upgrade those instances to Windows Server 2008. For more information, see Upgrading a Windows Server EC2 Instance to a Newer Version of Windows Server.

To find a Windows Server 2003 AMI

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. Choose **Owned by me**, and then choose **Public images**.
4. In the **Search** field, add the following filters and press Enter.
   a. Owner : Amazon images
   b. AMI Name : Windows_Server-2003

   Note
   The **Search** field is case sensitive.

Shared AMIs

A shared AMI is an AMI that a developer created and made available for other developers to use. One of the easiest ways to get started with Amazon EC2 is to use a shared AMI that has the components you need and then add custom content.

You use a shared AMI at your own risk. Amazon can't vouch for the integrity or security of AMIs shared by other Amazon EC2 users. Therefore, you should treat shared AMIs as you would any foreign code that you might consider deploying in your own data center and perform the appropriate due diligence.

We recommend that you get an AMI from a trusted source. If you have questions or observations about a shared AMI, use the AWS forums.

Amazon's public images have an aliased owner, which appears as `amazon` in the account field. This enables you to find AMIs from Amazon easily. Other users can't alias their AMIs.

Create the AMI. For more information, see Creating an Amazon EBS-Backed Windows AMI or Creating an Instance Store-Backed Windows AMI. For more information about building, delivering, and maintaining your applications on the AWS Marketplace, see the AWS Marketplace User Guide and AWS Marketplace Seller Guide.

Topics

- Finding Shared AMIs (p. 58)
- Making an AMI Public (p. 60)
- Sharing an AMI with Specific AWS Accounts (p. 62)
- Using Bookmarks (p. 63)
- Guidelines for Shared Windows AMIs (p. 64)

Finding Shared AMIs

You can use the Amazon EC2 console or the command line to find shared AMIs.

Finding a Shared AMI Using the Console

To find a shared private AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. In the first filter, select **Private images**. All AMIs that have been shared with you are listed. To granulate your search, click the Search bar and use the filter options provided in the menu.

To find a shared public AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. To find shared AMIs, select **Public images** from the **Filter** list. To granulate your search, click the Search bar and use the filter options provided in the menu.
4. Use filters to list only the types of AMIs that interest you. For example, select **Amazon images** to display only Amazon's public images.

Finding a Shared AMI Using the AWS CLI

To find a shared public AMI using the command line tools

Use the describe-images command to list AMIs. You can scope the list to the types of AMIs that interest you, as shown in the following examples.

The following command lists all public AMIs using the `--executable-users` option. This list includes any public AMIs that you own.

```
C:\> aws ec2 describe-images --executable-users all
```

The following command lists the AMIs for which you have explicit launch permissions. This list excludes any such AMIs that you own.

```
C:\> aws ec2 describe-images -executable-users self
```

The following command lists the AMIs owned by Amazon. Amazon's public AMIs have an aliased owner, which appears as `amazon` in the account field. This enables you to find AMIs from Amazon easily. Other users can't alias their AMIs.

```
C:\> aws ec2 describe-images --owners amazon
```

The following command lists the AMIs owned by the specified AWS account.

```
C:\> aws ec2 describe-images --owners 123456789012
```

To reduce the number of displayed AMIs, use a filter to list only the types of AMIs that interest you. For example, use the following filter to display only EBS-backed AMIs.

```
--filters "Name=root-device-type,Values=ebs"
```

Finding a Shared AMI Using the Amazon EC2 CLI

To find a shared public AMI using the command line tools

Use the ec2-describe-images command to list AMIs. You can scope the list to the types of AMIs that interest you, as shown in the following examples.

The following command lists all public AMIs using the *-x all* option. This list includes any public AMIs that you own.

```
C:\> ec2-describe-images -x all
```

The following command lists the AMIs for which you have explicit launch permissions. This list excludes any such AMIs that you own.

```
C:\> ec2-describe-images -x self
```

The following command lists the AMIs owned by Amazon. Amazon's public AMIs have an aliased owner, which appears as `amazon` in the account field. This enables you to find AMIs from Amazon easily. Other users can't alias their AMIs.

```
C:\> ec2-describe-images -o amazon
```

The following command lists the AMIs owned by the specified AWS account.

```
C:\> ec2-describe-images -o <target_uid>
```

The *<target_uid>* is the account ID that owns the AMIs for which you are looking.

To reduce the number of displayed AMIs, use a filter to list only the types of AMIs that interest you. For example, use the following filter to display only EBS-backed AMIs.

```
--filter "root-device-type=ebs"
```

Making an AMI Public

Amazon EC2 enables you to share your AMIs with other AWS accounts. You can allow all AWS accounts to launch the AMI (make the AMI public), or only allow a few specific accounts to launch the AMI. You are not billed when your AMI is launched by other AWS accounts; only the accounts launching the AMI are billed.

> **Note**
> If an AMI has a product code, you can't make it public. You must share the AMI with only specific AWS accounts.

Sharing a Public AMI Using the Console

To share a public AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. Select your AMI in the list, and then select **Modify Image Permissions** from the **Actions** list.

4. Select the **Public** radio button, and then click **Save**.

Sharing a Public AMI Using the AWS CLI

Each AMI has a `launchPermission` property that controls which AWS accounts, besides the owner's, are allowed to use that AMI to launch instances. By modifying the `launchPermission` property of an AMI, you can make the AMI public (which grants launch permissions to all AWS accounts) or share it with only the AWS accounts that you specify.

You can add or remove account IDs from the list of accounts that have launch permissions for an AMI. To make the AMI public, specify the `all` group. You can specify both public and explicit launch permissions.

To make an AMI public

Use the modify-image-attribute command as follows to add the `all` group to the `launchPermission` list for the specified AMI.

```
C:\> aws ec2 modify-image-attribute --image-id ami-2bb65342 --launch-permission
 "{\"Add\":[{\"Group\":\"all\"}]}"
```

To verify the launch permissions of the AMI, use the following describe-image-attribute command.

```
C:\> aws ec2 describe-image-attribute --image-id ami-2bb65342 --attribute
launchPermission
```

(Optional) To make the AMI private again, remove the `all` group from its launch permissions. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

```
C:\> aws ec2 modify-image-attribute --image-id ami-2bb65342 "{\"Re
move\":[{\"Group\":\"all\"}]}"
```

Sharing a Public AMI Using the Amazon EC2 CLI

Each AMI has a `launchPermission` property that controls which AWS accounts, besides the owner's, are allowed to use that AMI to launch instances. By modifying the `launchPermission` property of an AMI, you can make the AMI public (which grants launch permissions to all AWS accounts or share it with only the AWS accounts that you specify.

You can add or remove account IDs from the list of accounts that have launch permissions for an AMI To make the AMI public, specify the `all` group. You can specify both public and explicit launch permissions.

To make an AMI public

Use the ec2-modify-image-attribute command as follows to add the `all` group to the `launchPermission` list for the specified AMI.

```
C:\> ec2-modify-image-attribute ami-2bb65342  --launch-permission -a all
```

To verify the launch permissions of the AMI, use the following command.

```
C:\> ec2-describe-image-attribute ami-2bb65342 -l
```

To make the AMI private again, remove the `all` group from its launch permissions. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

```
C:\> ec2-modify-image-attribute ami-2bb65342 -l -r all
```

Sharing an AMI with Specific AWS Accounts

You can share an AMI with specific AWS accounts without making the AMI public. All you need are the AWS account IDs.

Sharing an AMI Using the Console

To grant explicit launch permissions using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. Select your AMI in the list, and then select **Modify Image Permissions** from the **Actions** list.
4. Specify the AWS account number of the user with whom you want to share the AMI in the **AWS Account Number** field, then click **Add Permission**.

   To share this AMI with multiple users, repeat the above step until you have added all the required users.
5. To allow create volume permissions for snapshots, check **Add "create volume" permissions to the following associated snapshots when creating permissions.**

   Note
   You do not need to share the Amazon EBS snapshots that an AMI references in order to share the AMI. Only the AMI itself needs to be shared; the system automatically provides the instance access to the referenced Amazon EBS snapshots for the launch.
6. Click **Save** when you are done.

Sharing an AMI Using the AWS CLI

Use the modify-image-attribute command to share an AMI as shown in the following examples.

To grant explicit launch permissions

The following command grants launch permissions for the specified AMI to the specified AWS account.

```
C:\> aws ec2 modify-image-attribute --image-id ami-2bb65342 --launch-permission
 "{\"Add\":[{\"UserId\":\"123456789012\"}]}"
```

To remove launch permissions for an account

The following command removes launch permissions for the specified AMI from the specified AWS account:

```
C:\> aws ec2 modify-image-attribute --image-id ami-2bb65342 "{\"Re
move\":[{\"UserId\":\"123456789012\"}]}"
```

To remove all launch permissions

The following command removes all public and explicit launch permissions from the specified AMI. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

```
C:\> aws ec2 reset-image-attribute --image-id ami-2bb65342 --attribute launch
Permission
```

Sharing an AMI Using the Amazon EC2 CLI

Use the ec2-modify-image-attribute command to share an AMI as shown in the following examples.

To grant explicit launch permissions

The following command grants launch permissions for the specified AMI to the specified AWS account.

```
C:\> ec2-modify-image-attribute ami-2bb65342 -l -a 111122223333
```

To remove launch permissions for an account

The following command removes launch permissions for the specified AMI from the specified AWS account:

```
C:\> ec2-modify-image-attribute ami-2bb65342 -l -r 111122223333
```

To remove all launch permissions

The following command removes all public and explicit launch permissions from the specified AMI. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

```
C:\> ec2-reset-image-attribute ami-2bb65342 -l
```

Using Bookmarks

If you have created a public AMI, or shared an AMI with another AWS user, you can create a *bookmark* that allows a user to access your AMI and launch an instance in their own account immediately. This is an easy way to share AMI references, so users don't have to spend time finding your AMI in order to use it.

Note that your AMI must be public, or you must have shared it with the user to whom you want to send the bookmark.

To create a bookmark for your AMI

1. Type a URL with the following information, where <*region*> is the region in which your AMI resides, and <*ami_id*> is the ID of the AMI:

   ```
   https://console.aws.amazon.com/ec2/v2/home?region=<region>#LaunchInstanceWiz
   ard:ami=<ami_id>
   ```

   For example, this URL launches an instance from the ami-2bb65342 AMI in the us-east-1 region:

   ```
   https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWiz
   ard:ami=ami-2bb65342
   ```

2. Distribute the link to users who want to use your AMI.
3. To use a bookmark, click the link or copy and paste it into your browser. The launch wizard opens, with the AMI already selected.

Guidelines for Shared Windows AMIs

Use the following guidelines to reduce the attack surface and improve the reliability of the AMIs you create.

Guidelines for creating AMIs

1. Launch and connect to a Windows instance. For more information, see Getting Started with Amazon EC2 Windows Instances (p. 20).
2. Customize the instance by installing the software and applications to share.
3. Do the following to make your AMI safe and reliable for sharing:
   - Develop a repeatable process for building, updating, and republishing AMIs.
   - Build AMIs using the most up-to-date operating systems, packages, and software.
   - Download and install the most recent version of the Amazon Windows Ec2Config Service.
   - Verify that Ec2SetPassword, Ec2WindowsActiviate and Ec2HandleUserData are enabled.
   - Verify that no guest accounts or Remote Desktop user accounts are present.
   - Disable or remove unnecessary services and programs to reduce the attack surface of your AMI.
   - Remove instance credentials, such as your key pair, from the AMI (if you saved them on the AMI). Store the credentials in a safe location.
   - Ensure that the administrator password and passwords on any other accounts are set to an appropriate value for sharing. These passwords are available for anyone who launches your shared AMI.
   - Test your AMI before you share it.
4. Create the AMI. For information about creating a shared AMI, see Creating an Amazon EBS-Backed Windows AMI (p. 68) or Creating an Instance Store-Backed Windows AMI (p. 70). For more information about building, delivering, and maintaining your applications on the AWS Marketplace, see the AWS Marketplace Seller Guide.

Paid AMIs

A *paid AMI* is an AMI that you can purchase from a developer.

Amazon EC2 integrates with AWS Marketplace, enabling developers to charge other Amazon EC2 users for the use of their AMIs or to provide support for instances.

The AWS Marketplace is an online store where you can buy software that runs on AWS; including AMIs that you can use to launch your EC2 instance. The AWS Marketplace AMIs are organized into categories, such as Developer Tools, to enable you to find products to suit your requirements. For more information about AWS Marketplace, see the AWS Marketplace site.

Launching an instance from a paid AMI is the same as launching an instance from any other AMI. No additional parameters are required. The instance is charged according to the rates set by the owner of the AMI, as well as the standard usage fees for the related web services; for example, the hourly rate for running a m1.small instance type in Amazon EC2. The owner of the paid AMI can confirm whether a specific instance was launched using that paid AMI.

Important
Amazon DevPay is no longer accepting new sellers or products. AWS Marketplace is now the single, unified e-commerce platform for selling software and services through AWS. For information about how to deploy and sell software from AWS Marketplace, see Selling on AWS Marketplace. AWS Marketplace supports AMIs backed by Amazon EBS.

Topics

- Selling Your AMI (p. 65)
- Finding a Paid AMI (p. 65)
- Purchase a Paid AMI (p. 66)
- Getting the Product Code for Your Instance (p. 67)
- Using Paid Support (p. 67)
- Bills for Paid and Supported AMIs (p. 67)
- Managing Your AWS Marketplace Subscriptions (p. 68)

Selling Your AMI

You can sell your AMI using AWS Marketplace. AWS Marketplace offers an organized shopping experience. Additionally, AWS Marketplace also supports AWS features such as Amazon EBS-backed AMIs, Reserved Instances, and Spot instances.

For information about how to sell your AMI on AWS Marketplace, see Selling on AWS Marketplace.

Finding a Paid AMI

There are several ways that you can find AMIs that are available for you to purchase. For example, you can use AWS Marketplace, the Amazon EC2 console, or the command line. Alternatively, a developer might let you know about a paid AMI themselves.

Finding a Paid AMI Using the Console

To find a paid AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. Select **Public images** from the first **Filter** list. Click the Search bar and select **Product Code**, then **Marketplace**. Click the Search bar again, select **Platform** and then choose the operating system from the list.

Finding a Paid AMI Using AWS Marketplace

To find a paid AMI using AWS Marketplace

1. Open AWS Marketplace.
2. Enter the name of the operating system in the search box, and click **Go**.
3. To scope the results further, use one of the categories or filters.
4. Each product is labeled with its product type: either `AMI` or `Software as a Service`.

Finding a Paid AMI Using the AWS CLI

You can find a paid AMI using the describe-images command as follows.

```
C:\> ec2-describe-images --owners aws-marketplace
```

This command returns numerous details that describe each AMI, including the product code for a paid AMI. The output from `describe-images` includes an entry for the product code like the following:

```
"ProductCodes": [
    {
        "ProductCodeId": "product_code",
        "ProductCodeType": "marketplace"
    }
],
```

Finding a Paid AMI Using the Amazon EC2 CLI

You can find a paid AMI using the ec2-describe-images command as follows.

```
C:\> ec2-describe-images -o aws-marketplace
```

This command returns numerous details that describe each AMI, including the product code for a paid AMI. The following example output from `ec2-describe-images` includes a product code.

```
IMAGE   ami-a5bf59cc    image_source    123456789012    available public
product_code        x86_64    machine                     instance-store
```

Purchase a Paid AMI

You must sign up for (purchase) a paid AMI before you can launch an instance using the AMI.

Typically a seller of a paid AMI presents you with information about the AMI, including its price and a link where you can buy it. When you click the link, you're first asked to log into AWS, and then you can purchase the AMI.

Purchasing a Paid AMI Using the Console

You can purchase a paid AMI by using the Amazon EC2 launch wizard. For more information, see Launching an AWS Marketplace Instance (p. 213).

Subscribing to a Product Using AWS Marketplace

To use the AWS Marketplace, you must have an AWS account. To launch instances from AWS Marketplace products, you must be signed up to use the Amazon EC2 service, and you must be subscribed to the product from which to launch the instance. There are two ways to subscribe to products in the AWS Marketplace:

- **AWS Marketplace website**: You can launch preconfigured software quickly with the 1-Click deployment feature.
- **Amazon EC2 launch wizard**: You can search for an AMI and launch an instance directly from the wizard. For more information, see Launching an AWS Marketplace Instance (p. 213).

Purchasing a Paid AMI From a Developer

The developer of a paid AMI can enable you to purchase a paid AMI that isn't listed in AWS Marketplace. The developer provides you with a link that enables you to purchase the product through Amazon. You can sign in with your Amazon.com credentials and select a credit card that's stored in your Amazon.com account to use when purchasing the AMI.

Getting the Product Code for Your Instance

You can retrieve the AWS Marketplace product code for your instance using its instance metadata. For more information about retrieving metadata, see Instance Metadata and User Data (p. 160).

To retrieve a product code, use the following query:

```
C:\> GET http://169.254.169.254/latest/meta-data/product-codes
```

If the instance has a product code, Amazon EC2 returns it. For example:

```
774F4FF8
```

Using Paid Support

Amazon EC2 also enables developers to offer support for software (or derived AMIs). Developers can create support products that you can sign up to use. During sign-up for the support product, the developer gives you a product code, which you must then associate with your own AMI. This enables the developer to confirm that your instance is eligible for support. It also ensures that when you run instances of the product, you are charged according to the terms for the product specified by the developer.

> **Important**
> You can't use a support product with Reserved Instances. You always pay the price that's specified by the seller of the support product.

To associate a product code with your AMI, use one of the following commands, where *ami_id* is the ID of the AMI and *product_code* is the product code:

- modify-image-attribute (AWS CLI)

```
C:\> aws ec2 modify-image-attribute --image-id ami_id --product-codes
"product_code"
```

- ec2-modify-image-attribute (Amazon EC2 CLI)

```
C:\> ec2-modify-image-attribute ami_id --product-code product_code
```

After you set the product code attribute, it cannot be changed or removed.

Bills for Paid and Supported AMIs

At the end of each month, you receive an email with the amount your credit card has been charged for using any paid or supported AMIs during the month. This bill is separate from your regular Amazon EC2 bill. For more information, see Paying For AWS Marketplace Products.

Managing Your AWS Marketplace Subscriptions

On the AWS Marketplace website, you can check your subscription details, view the vendor's usage instructions, manage your subscriptions, and more.

To check your subscription details

1. Log in to the AWS Marketplace.
2. Click **Your Account**.
3. Click **Manage Your Software Subscriptions**.
4. All your current subscriptions are listed. Click **Usage Instructions** to view specific instructions for using the product, for example, a user name for connecting to your running instance.

To cancel an AWS Marketplace subscription

1. Ensure that you have terminated any instances running from the subscription.
   a. Open the Amazon EC2 console.
   b. In the navigation pane, click **Instances**.
   c. Select the instance, click **Actions**, select **Instance State**, and select **Terminate**. When prompted, click **Yes, Terminate**.
2. Log in to the AWS Marketplace, and click **Your Account**, then **Manage Your Software Subscriptions**.
3. Click **Cancel subscription**. You are prompted to confirm your cancellation.

   Note
   After you've canceled your subscription, you are no longer able to launch any instances from that AMI. To use that AMI again, you need to resubscribe to it, either on the AWS Marketplace website, or through the launch wizard in the Amazon EC2 console.

Creating an Amazon EBS-Backed Windows AMI

To create an Amazon EBS-backed Windows AMI, you launch and customize a Windows instance, then you create the AMI.

If you need to create an Amazon EBS-backed Linux AMI, see Creating an Amazon EBS-Backed Linux AMI in the *Amazon EC2 User Guide for Linux Instances*.

The AMI creation process is different for instance store-backed AMIs. For more information about the differences between Amazon EBS-backed and instance store-backed instances, and how to determine the root device type for your instance, see Root Device Volume (p. 8). If you need to create an instance store-backed Windows AMI, see Creating an Instance Store-Backed Windows AMI (p. 70).

Creating an AMI from an Instance

To create an Amazon EBS-backed AMI from an instance using the console

1. If you don't have a running instance that uses an Amazon EBS volume for the root device, you must launch one.
   a. Open the Amazon EC2 console.

b. In the navigation pane, click **AMIs**. Select an Amazon EBS-backed AMI that is similar to the AMI that you want to create. To view the Amazon EBS-backed Windows AMIs, select the following options from the **Filter** lists: **Public images**, **EBS images**, and then **Windows**.

   You can select any public AMI that uses the version of Windows Server that you want for your AMI. However, you must select an Amazon EBS-backed AMI; don't start with an instance store-backed AMI.

c. Click **Launch** to launch an instance of the Amazon EBS-backed AMI that you've selected. Accept the default values as you step through the wizard.

2. While the instance is running, connect to it and customize it. For example, you can perform any of the following actions on your instance:

   - Install software and applications.
   - Copy data.
   - Reduce start time by deleting temporary files, defragmenting your hard drive, and zeroing out free space.
   - Create a new user account and add it to the Administrators group.

     Tip
     If you are sharing your AMI, these credentials can be supplied for RDP access without disclosing your default Administrator password.

   - Configure settings using EC2Config. **If you want your AMI to generate a random password at launch time, you need to enable the `Ec2SetPassword` plugin; otherwise, the current Administrator password is used.** For more information, see Configuring a Windows Instance Using the EC2Config Service (p. 235).

3. If the instance uses RedHat drivers to access Xen virtualized hardware, upgrade to Citrix drivers before you create an AMI. For more information, see Upgrading PV Drivers on Your Windows AMI (p. 265).
4. (Optional) When the instance is set up the way you want it, it is best to stop the instance before you create the AMI, to ensure data integrity. You can use EC2Config to stop the instance, or select the instance in the Amazon EC2 console, click **Actions**, select **Instance State**, and then click **Stop**.
5. On the **Instances** page of the Amazon EC2 console, select your instance. Click **Actions**, select **Image**, and then click **Create Image**.

   Tip
   If this option is disabled, your instance isn't an Amazon EBS-backed instance.

6. In the **Create Image** dialog box, specify a unique name and an optional description for the AMI (up to 255 characters).
7. To add an Amazon EBS volume, click **Add New Volume**, and select `EBS` from the **Type** list. Fill in the other information as required.

   When you launch an instance from your new AMI, these additional volumes are automatically attached to the instance. Empty volumes must be formatted and mounted. Volumes based on a snapshot must be mounted.
8. To add an instance store volume, click **Add New Volume**, and select `Instance Store` from the **Type** list. Fill in the other information as required.

   When you launch an instance from your new AMI, these additional volumes are automatically initialized and mounted. These volumes don't contain data from the instance store volumes of the running instance from which you based your AMI.
9. Click **Create Image** to start creating the AMI.

To view the status of your AMI, go to the **AMIs** page. While your AMI is being created, its status is `pending`. It takes a few minutes to complete the AMI creation process. When the process has completed, the status of your AMI is `available`. If you go to the **Snapshots** page, you'll see that we created a snapshot that's used to create the root device volume of any instance that you launch using your new AMI.

When you are ready to delete your AMI and snapshot, see Deregistering Your AMI (p. 76).

To create an Amazon EBS-backed AMI from an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- create-image (AWS CLI)
- ec2-create-image (Amazon EC2 CLI)
- New-EC2Image (AWS Tools for Windows PowerShell)

Creating an Instance Store-Backed Windows AMI

To create an instance store-backed Windows AMI, first launch and customize a Windows instance, then bundle the instance, and register an AMI from the manifest that's created during the bundling process.

> **Important**
> The only Windows AMIs that can be backed by instance store are those for Windows Server 2003. Instance store-backed instances don't have the available disk space required for later versions of Windows Server.

You can only bundle an instance store-backed Windows instance using this procedure. If you need to create an instance store-backed Linux AMI, see Creating an Instance Store-Backed Linux AMI in the *Amazon EC2 User Guide for Linux Instances*.

The AMI creation process is different for Amazon EBS-backed AMIs. For more information about the differences between Amazon EBS-backed and instance store-backed instances, and how to determine the root device type for your instance, see Root Device Volume (p. 8). If you need to create an Amazon EBS-backed Windows AMI, see Creating an Amazon EBS-Backed Windows AMI (p. 68).

Contents

- Instance Store-Backed Windows AMIs (p. 70)
- Preparing to Create an Instance Store-Backed Windows AMI (p. 71)
- Bundling an Instance Store-Backed Windows Instance (p. 72)
- Registering an Instance Store-Backed Windows AMI (p. 73)

Instance Store-Backed Windows AMIs

Instances launched from an AMI backed by instance store use an instance store volume as the root device volume. The image of the root device volume of an instance store-backed AMI is initially stored in Amazon S3. When an instance is launched using an instance store-backed AMI, the image of its root device volume is copied from Amazon S3 to the root partition of the instance. The root device volume is then used to boot the instance.

When you create an instance store-backed AMI, it must be uploaded to Amazon S3. Amazon S3 stores data objects in buckets, which are similar in concept to directories. Buckets have globally unique names and are owned by unique AWS accounts.

Bundling Process

The bundling process comprises the following tasks:

- Compress the image to minimize bandwidth usage and storage requirements.
- Encrypt and sign the compressed image to ensure confidentiality and authenticate the image against its creator.
- Split the encrypted image into manageable parts for upload.
- Run `Sysprep` to strip computer-specific information (for example, the MAC address and computer name) from the Windows AMI to prepare it for virtualization.
- Create a manifest file that contains a list of the image parts with their checksums.
- Put all components of the AMI in the Amazon S3 bucket that you specify when making the bundle request.

Storage Volumes

It is important to remember the following details about the storage for your instance when you create an instance store-backed AMI:

- The root device volume (C:) is automatically attached when a new instance is launched from your new AMI. The data on any other instance store volumes is deleted when the instance is bundled.
- The instance store volumes other than the root device volume (for example, D:) are temporary and should be used only for short-term storage.
- You can add Amazon EBS volumes to your instance store-based instance. Amazon EBS volumes are stored within Amazon S3 buckets and remain intact when the instance is bundled. Therefore, we recommend that you store all the data that must persist on Amazon EBS volumes, not instance store volumes.

For more information about Amazon EC2 storage options, see Storage (p. 515).

Preparing to Create an Instance Store-Backed Windows AMI

When you create an AMI, you start by basing it on an instance. You can customize the instance to include the data and software that you need. As a result, any instance that you launch from your AMI has everything that you need.

To launch an instance store-backed Windows instance

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**. Select an instance store-backed AMI that is similar to the AMI that you want to create. To view the instance store-backed Windows AMIs, select the following options from the **Filter** lists: **Public images**, **Instance store images**, and then **Windows**.

   You can select any public AMI that uses the version of Windows Server that you want for your AMI. However, you must select an instance store-backed AMI; don't start with an Amazon EBS-backed AMI.

3. Click **Launch** to launch an instance of the instance store-backed AMI that you've selected. Accept the default values as you step through the wizard.
4. While the instance is running, connect to it and customize it. For example, you can perform any of the following on your instance:

   - Install software and applications.
   - Copy data.
   - Reduce start time by deleting temporary files, defragmenting your hard drive, and zeroing out free space.
   - Create a new user account and add it to the Administrators group.

     Tip
     If you are sharing your AMI, these credentials can be provided for RDP access without disclosing your default Administrator password.

   - Configure settings using EC2Config. For example, to generate a random password for your instance when you launch it from this AMI, enable the Ec2SetPassword plugin; otherwise, the current Administrator password is used. For more information, see Configuring a Windows Instance Using the EC2Config Service (p. 235).

5. If the instance uses RedHat drivers to access Xen virtualized hardware, upgrade to Citrix drivers before you create an AMI. For more information, see Upgrading PV Drivers on Your Windows AMI (p. 265).

Bundling an Instance Store-Backed Windows Instance

Now that you've customized your instance, you can bundle the instance to create an AMI, using either the AWS Management Console or the command line.

To bundle an instance store-backed Windows instance using the console

1. Determine whether you'll use an existing Amazon S3 bucket for your new AMI or create a new one. To create a new Amazon S3 bucket, use the following steps:

   a. Open the Amazon S3 console.
   b. Click **Create Bucket**.
   c. Specify a name for the bucket and click **Create**.

2. Open the Amazon EC2 console.
3. In the navigation pane, click **Instances**. Right-click the instance you set up in the previous procedure, and select **Bundle Instance (instance store AMI)**.
4. In the **Bundle Instance** dialog box, fill in the requested information, and then click **OK**:

   - **Amazon S3 bucket name**: Specify the name of an S3 bucket that you own. The bundle files and manifest will be stored in this bucket.
   - **Amazon S3 key name**: Specify a prefix for the files that are generated by the bundle process.

   The **Bundle Instance** dialog box confirms that the request to bundle the instance has succeeded, and also provides the ID of the bundle task. Click **Close**.

To view the status of the bundle task, click **Bundle Tasks** in the navigation pane. The bundle task progresses through several states, including `waiting-for-shutdown`, `bundling`, and `storing`. If the bundle task can't be completed successfully, the status is `failed`.

To bundle an instance store-backed Windows instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- bundle-instance (AWS CLI)
- ec2-bundle-instance (Amazon EC2 CLI)
- New-EC2InstanceBundle (AWS Tools for Windows PowerShell)

Registering an Instance Store-Backed Windows AMI

Finally, you must register your AMI so that Amazon EC2 can locate it and launch instances from it.

Your new AMI is stored in Amazon S3. You'll incur charges for this storage until you deregister the AMI and delete the bundle in Amazon S3.

If you make any changes to the source AMI stored in Amazon S3, you must deregister and reregister the AMI before the changes take effect.

To register an instance store-backed Windows AMI from the AMI page in the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**. By default, the console displays the AMIs that you own.
3. Click **Actions** and select **Register new AMI**.
4. In the **Register Image** dialog box, provide the **AMI Manifest Path** and then click **Register**.

To register an instance store-backed Windows AMI from the Bundle Tasks page in the console

1. On the navigation pane, click **Bundle Tasks**.
2. Select the bundle task, and click **Register as an AMI**.
3. A dialog displays the AMI manifest path. Click **Register**, and then click **Close** in the confirmation dialog box.

To register an instance store-backed Windows AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- register-image (AWS CLI)
- ec2-register (Amazon EC2 CLI)
- Register-EC2Image (AWS Tools for Windows PowerShell)

To view your new AMI, click **AMIs** in the navigation pane, and ensure the **Owned by me** filter option is selected.

Copying an AMI

You can easily copy the Amazon Machine Images (AMIs) that you own to other AWS regions and scale your applications to take advantage of AWS's geographically diverse regions.

Copying your AMIs provides the following benefits:

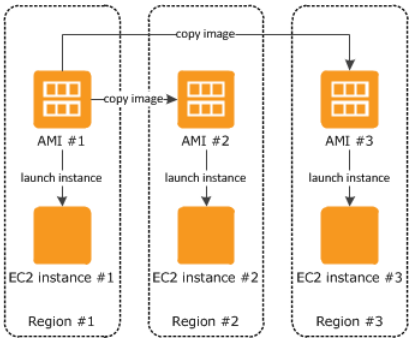
- Consistent global deployment: You can copy an AMI from one region to another, enabling you to launch consistent instances based from the same AMI into different regions.
- Scalability: You can more easily design and build world-scale applications that meet the needs of your users, regardless of their location.
- Performance: You can increase performance by distributing your application, as well as locating critical components of your application in closer proximity to your users. You can also take advantage of region-specific features, such as instance types or other AWS services.
- High availability: You can design and deploy applications across AWS regions, to increase availability.

There are no charges for copying an AMI. However, standard storage and data transfer rates apply.

> **Note**
> Destination regions are limited to 50 concurrent AMI copies at a time, with no more than 25 of those coming from a single source region. To request an increase to this limit, see Amazon EC2 Service Limits (p. 618).

AMI Copy

You can copy both Amazon EBS-backed AMIs and instance store-backed AMIs. You can copy an AMI to as many regions as you like. You can also copy an AMI to the same region. Each copy of an AMI results in a new AMI with its own unique AMI ID. When you launch an instance from an AMI, we launch it into the same region as the AMI you select, as shown in the following diagram.



When you copy an AMI, the new AMI is fully independent of the source AMI; there is no link to the original (source) AMI. You can modify the new AMI without affecting the source AMI. The reverse is also true: you can modify the source AMI without affecting the new AMI. Therefore, if you make changes to the source AMI and want those changes to be reflected in the AMI in the destination region, you must recopy the source AMI to the destination region.

We don't copy launch permissions, user-defined tags, or Amazon S3 bucket permissions from the source AMI to the new AMI. After the copy operation is complete, you can apply launch permissions, user-defined tags, and Amazon S3 bucket permissions to the new AMI. AMIs with encrypted volumes cannot be copied.

When you first copy an instance store-backed AMI to a region, we create an Amazon S3 bucket for the AMIs copied to that region. All instance store-backed AMIs that you copy to that region are stored in this

bucket. The names of these buckets have the following format: amis-for-*account*-in-*region*-*hash*. For example: `amis-for-123456789012-in-us-west-2-yhjmxvp6`.

Copying an Amazon EC2 AMI

You can copy an AMI using the AWS Management Console or the command line.

Important
If you have a Windows AMI with encrypted volumes, you can't copy it.

Prior to copying an AMI, you must ensure that the contents of the source AMI are updated to support running in a different region. For example, you should update any database connection strings or similar application configuration data to point to the appropriate resources. Otherwise, instances launched from the new AMI in the destination region may still use the resources from the source region, which can impact performance and cost.

To copy an AMI using the console

1. Open the Amazon EC2 console.
2. From the navigation bar, select the region that contains the AMI to copy.
3. In the navigation pane, click **AMIs**.
4. Select the AMI to copy, click **Actions**, and then click **Copy AMI**.
5. In the **AMI Copy** page, set the following fields, and then click **Copy AMI**:
   - **Destination region**: Select the region to which you want to copy the AMI.
   - **Name**: Specify a name for the new AMI.
   - **Description**: By default, the description includes information about the source AMI so that you can identify a copy from the original. You can change this description as necessary.
6. We display a confirmation page to let you know that the copy operation has been initiated and provide you with the ID of the new AMI.

   To check on the progress of the copy operation immediately, click the provided link to switch to the destination region. To check on the progress later, click **Done**, and then when you are ready, use the navigation pane to switch to the destination region.

   The initial status of the destination AMI is `pending` and the operation is complete when the status is `available`.

To copy an AMI using the command line

Copying an AMI from the command line requires that you specify both the source and destination regions. You specify the source region using the `--source-region` parameter. For the destination region, you have two options:

- Use the `--region` parameter.
- Set an environmental variable. For more information, see Setting Up the CLI Tools (Windows).

You can copy an AMI using one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- copy-image (AWS CLI)
- ec2-copy-image (Amazon EC2 CLI)
- Copy-EC2Image (AWS Tools for Windows PowerShell)

Stopping a Pending AMI Copy Operation

You can stop a pending AMI copy using the AWS Management Console or the command line.

To stop an AMI copy operation using the console

1. Open the Amazon EC2 console.
2. From the navigation bar, select the destination region from the region selector.
3. In the navigation pane, click **AMIs**.
4. Select the AMI you want to stop copying, click **Actions**, and then click **Deregister**.
5. When asked for confirmation, click **Continue**.

To stop an AMI copy operation using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- deregister-image (AWS CLI)
- ec2-deregister (Amazon EC2 CLI)
- Unregister-EC2Image (AWS Tools for Windows PowerShell)

Deregistering Your AMI

You can deregister an AMI when you have finished using it. After you deregister an AMI, you can't use it to launch new instances.

When you deregister an AMI, it doesn't affect any instances that you've already launched from the AMI. You'll continue to incur usage costs for these instances. Therefore, if you are finished with these instances, you should terminate them.

The procedure that you'll use to clean up your AMI depends on whether it is backed by Amazon EBS or instance store. (Note that the only Windows AMIs that can be backed by instance store are those for Windows Server 2003.)

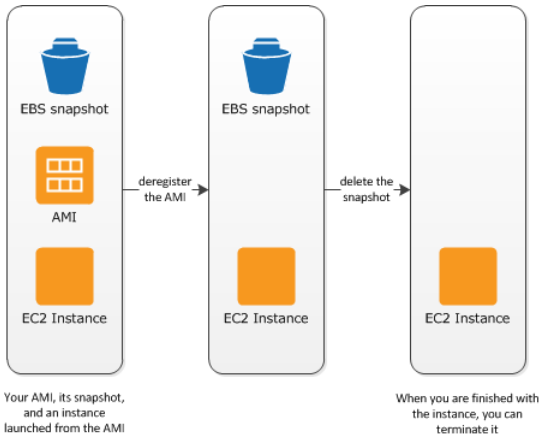
Contents

- Cleaning Up Your Amazon EBS-Backed AMI (p. 76)
- Cleaning Up Your Instance Store-Backed AMI (p. 77)

Cleaning Up Your Amazon EBS-Backed AMI

When you deregister an Amazon EBS-backed AMI, it doesn't affect the snapshot that was created for the root volume of the instance during the AMI creation process. You'll continue to incur storage costs for this snapshot. Therefore, if you are finished with the snapshot, you should delete it.

The following diagram illustrates the process for cleaning up your Amazon EBS-backed AMI.



To clean up your Amazon EBS-backed AMI

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**. Select the AMI, click **Actions**, and then click **Deregister**. When prompted for confirmation, click **Continue**.

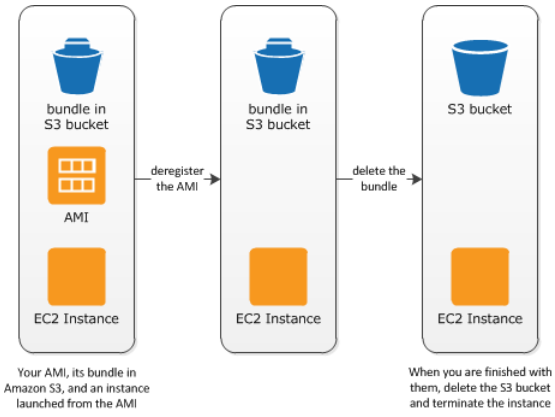
   The AMI status is now `unavailable`.

   Note
   It may take a few minutes before the console changes the status from `available` to `unavailable`, or removes the AMI from the list altogether. Click the **Refresh** button to refresh the status.
3. In the navigation pane, click **Snapshots**. Select the snapshot and click **Delete Snapshot**. When prompted for confirmation, click **Yes, Delete**.
4. (Optional) If you are finished with an instance that you launched from the AMI, terminate it. In the navigation pane, click **Instances**. Select the instance, click **Actions**, and then click **Terminate**. When prompted for confirmation, click **Yes, Terminate**.

Cleaning Up Your Instance Store-Backed AMI

When you deregister an instance store-backed AMI, it doesn't affect the files that you uploaded to Amazon S3 when you created the AMI. You'll continue to incur usage costs for these files in Amazon S3. Therefore, if you are finished with these files, you should delete them.

The following diagram illustrates the process for cleaning up your instance store-backed AMI.



To clean up your instance store-backed AMI

1. Deregister the AMI using the ec2-deregister command as follows.

```
ec2-deregister ami_id
```

   The AMI status is now `unavailable`.
2. Delete the bundle using the ec2-delete-bundle command as follows.

```
ec2-delete-bundle -b myawsbucket/myami -a your_access_key_id -s
your_secret_access_key -p image
```

3. (Optional) If you are finished with an instance that you launched from the AMI, you can terminate it using the ec2-terminate-instances command as follows.

```
ec2-terminate-instances instance_id
```

4. (Optional) If you are finished with the Amazon S3 bucket that you uploaded the bundle to, you can delete the bucket. To delete an Amazon S3 bucket, open the Amazon S3 console, select the bucket, click **Actions**, and then click **Delete**.

AWS Windows AMI Version History

AWS provides Amazon Machine Images (AMIs) that contain versions of Microsoft Windows Server, known as the *AWS Windows AMIs*. Some AWS Windows AMIs also come configured with Microsoft SQL Server or Internet Information Services (IIS). You can use an AMI with Microsoft SQL Server and IIS already configured, or you can start from a basic Windows AMI, and then install Microsoft SQL Server and enable IIS on the instance. For more information, see AWS Windows AMIs (p. 51).

Contents

- Configuration Settings and Drivers (p. 78)
- Updating Your Windows Instance (p. 79)
- Determining Your Instance Version (p. 79)
- AWS Windows AMI Versions (p. 80)
- Image Changes (p. 88)
- Subscribing to Windows AMI Notifications (p. 89)
- Upgrading or Migrating a Windows Server Instance (p. 90)

Configuration Settings and Drivers

The AWS Windows AMIs are generally configured the same way as a Windows Server that you install from Microsoft-issued media. There are, however, a few differences in the installation defaults.

AWS Windows AMIs come with an additional service installed, the EC2Config service. The EC2Config service runs in the local system account and is primarily used during the initial setup. For information about the tasks that EC2Config performs, see Overview of EC2Config Tasks (p. 236).

After you launch your Windows instance with its initial configuration, you can use the EC2Config service to change the configuration settings as part of the process of customizing and creating your own AMI. Instances launched from your customized AMI are launched with the new configuration.

AWS Windows AMIs contain a set of drivers to permit access to Xen virtualized hardware. These drivers are used by Amazon EC2 to map instance store and Amazon EBS volumes to their devices. For more information, see Paravirtual Drivers (p. 258).

Updating Your Windows Instance

After you launch a Windows instance, you are responsible for installing updates on it. You can manually install only the updates that interest you, or you can start from a current AWS Windows AMI and build a new Windows instance. For information about finding the current AWS Windows AMIs, see Finding a Windows AMI (p. 56).

For Windows instances, you can install updates to the following services or applications:

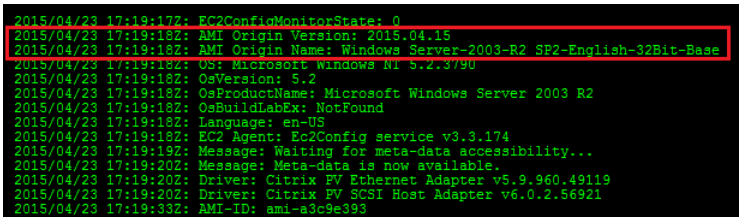
- Windows
- Microsoft SQL Server
- Windows PowerShell
- EC2Config service (p. 256)
- PV Drivers (p. 265)
- AWS Tools for Windows PowerShell
- AWS CloudFormation helper scripts

You can reboot a Windows instance after installing updates. For more information, see Reboot Your Instance (p. 222).

Determining Your Instance Version

The AWS Management Console provides details about the AMI that you use to create an Amazon EC2 instance. The **AMI ID** field on the **Description** tab contains information including the Windows Server SKU, the architecture (32-bit or 64-bit), the date the AMI was created, and an AMI ID.

If an AMI has been made private or replaced by later versions and is no longer listed in the catalog, the **AMI ID** field states, "Cannot load detail for ami-xxxxx. You may not be permitted to view it." To determine which AMI was used to create the instance, you must open the system log. In the EC2 console, choose an instance, and from the context-menu (right-click) choose **Instance Settings** and then choose **Get System Log**. The date the AMI was created and the SKU are listed in the **AMI Origin Version** and **AMI Origin Name** fields.



Note
The **AMI Origin Version** and **AMI Origin Name** are displayed in the system log only if the EC2Config service is running version 2.1.19 or later and the AMI was created after 2013.11.13.

AWS Windows AMI Versions

AWS provides updated, fully-patched Windows AMIs within five business days of Microsoft's patch Tuesday (the second Tuesday of each month). The new AMIs are available immediately through the **Images** page in the Amazon EC2 console. The new AMIs are available in the AWS Marketplace and the **Quick Start** tab of the launch instance wizard within a few days of their release. AWS makes the previously published Windows AMIs private within 10 business days after publishing updated Windows AMIs, to ensure that customers have the latest security updates by default.

The Windows AMIs in each release have new AMI IDs. Therefore, we recommend that you write scripts that locate the latest AWS Windows AMIs by their names, rather than by their IDs. For more information, see Get-EC2ImageByName in the *AWS Tools for Windows PowerShell User Guide*. You can also create a Lambda function to perform this task with Amazon EC2 and other services such as AWS CloudFormation. For more information, see Create a Lambda Function.

The following table summarizes the changes to each release of the AWS Windows AMIs. Note that some changes apply to all AWS Windows AMIs while others apply to only a subset of these AMIs.

Release	Changes
2015.9.9	**ALL AMIs** • Microsoft security updates current to September 2015 • EC2Config service version 3.9.359 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation helper scripts
2015.8.18	**ALL AMIs** • Microsoft security updates current to August 2015 • EC2Config service version 3.8.294 • Current AWS Tools for Windows PowerShell **Only AMIs with Windows Server 2012 and Windows Server 2012 R2** • AWS PV Driver 7.3.2
2015.7.21	**ALL AMIs** • Microsoft security updates current to July 2015 • EC2Config service version 3.7.308 • Current AWS Tools for Windows PowerShell • Modified AMI descriptions of SQL images for consistency

Release	Changes
2015.6.10	**ALL AMIs** • Microsoft security updates current to June 2015 • EC2Config service version 3.6.269 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation helper scripts **Only AMIs with Windows Server 2012 R2** • AWS PV Driver 7.3.1
2015.5.13	**All AMIs** • Microsoft security updates current to May 2015 • EC2Config service version 3.5.228 • Current AWS Tools for Windows PowerShell
2015.04.15	**All AMIs** • Microsoft security updates current to April 2015 • EC2Config service version 3.3.174 • Current AWS Tools for Windows PowerShell
2015.03.11	**All AMIs** • Microsoft security updates current to March 2015 • EC2Config service version 3.2.97 • Current AWS Tools for Windows PowerShell **Only AMIs with Windows Server 2012 R2** • AWS PV Driver 7.3.0
2015.02.11	**All AMIs** • Microsoft security updates current to February 2015 • EC2Config service version 3.0.54 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation helper scripts

Release	Changes
2015.01.14	**All AMIs** • Microsoft security updates current to January 2015 • EC2Config service version 2.3.313 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation helper scripts
2014.12.10	**All AMIs** • Microsoft security updates current to December 2014 • EC2Config service version 2.2.12 • Current AWS Tools for Windows PowerShell
2014.11.19	**All AMIs** • Microsoft security updates current to November 2014 • EC2Config service version 2.2.11 • Current AWS Tools for Windows PowerShell
2014.10.15	**All AMIs** • Microsoft security updates current to October 2014 • EC2Config service version 2.2.10 • Current AWS Tools for Windows PowerShell **Only AMIs with Windows Server 2012 R2** • AWS PV Driver 7.2.4.1 (resolves the issues with Plug and Play Cleanup, which is now enabled by default)
2014.09.10	**All AMIs** • Microsoft security updates current to September 2014 • EC2Config service version 2.2.8 • Current AWS Tools for Windows PowerShell **Only AMIs with Windows Server 2012 R2** • Disable Plug and Play Cleanup (see Important information) • AWS PV Driver 7.2.2.1 (resolves issues with the uninstaller)

Release	Changes
2014.08.13	**All AMIs** • Microsoft security updates current to August 2014 • EC2Config service version 2.2.7 • Current AWS Tools for Windows PowerShell **Only AMIs with Windows Server 2012 R2** • AWS PV Driver 7.2.2.1 (improves disk performance, resolves issues with reconnecting multiple network interfaces and lost network settings)
2014.07.10	**All AMIs** • Microsoft security updates current to July 2014 • EC2Config service version 2.2.5 • Current AWS Tools for Windows PowerShell
2014.06.12	**All AMIs** • Microsoft security updates current to June 2014 • EC2Config service version 2.2.4 • Removed NVIDIA drivers (except for Windows Server 2012 R2 AMIs) • Current AWS Tools for Windows PowerShell
2014.05.14	**All AMIs** • Microsoft security updates current to May 2014 • EC2Config service version 2.2.2 • Current AWS Tools for Windows PowerShell • AWS CloudFormation helper scripts version 1.4.0
2014.04.09	**All AMIs** • Microsoft security updates current to April 2014 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation helper scripts
2014.03.12	**All AMIs** • Microsoft security updates current to March 2014

Release	Changes
2014.02.12	**All AMIs** • Microsoft security updates current to February 2014 • EC2Config service version 2.2.1 • Current AWS Tools for Windows PowerShell • KB2634328 • Remove the BCDEdit `useplatformclock` value **Only AMIs with Microsoft SQL Server** • Microsoft SQL Server 2012 SP1 cumulative update package 8 • Microsoft SQL Server 2008 R2 cumulative update package 10
2013.11.13	**All AMIs** • Microsoft security updates current to November 2013 • EC2Config service version 2.1.19 • Current AWS Tools for Windows PowerShell • Configure NTP to synchronize the time once a day (the default is every seven days) **Only AMIs with Windows Server 2012** • Clean up the WinSXS folder using the following command: `dism /online /cleanup-image /StartComponentCleanup`
2013.09.11	**All AMIs** • Microsoft security updates current to September 2013 • EC2Config service version 2.1.18 • Current AWS Tools for Windows PowerShell • AWS CloudFormation helper scripts version 1.3.15
2013.07.10	**All AMIs** • Microsoft security updates current to July 2013 • EC2Config service version 2.1.16 • Expanded the root volume to 50 GB • Set the page file to 512 MB, expanding to 8 GB as needed • Current AWS Tools for Windows PowerShell

Release	Changes
2013.06.12	**All AMIs** • Microsoft security updates current to June 2013 • Current AWS Tools for Windows PowerShell **Only AMIs with Microsoft SQL Server** • Microsoft SQL Server 2012 SP1 with cumulative update package 4
2013.05.15	**All AMIs** • Microsoft security updates current to May 2013 • EC2Config service version 2.1.15 • All instance store volumes attached by default • Remote PowerShell enabled by default • Current AWS Tools for Windows PowerShell
2013.04.14	**All AMIs** • Microsoft security updates current to April 2013 • Current AWS Tools for Windows PowerShell • AWS CloudFormation helper scripts version 1.3.14
2013.03.14	**All AMIs** • Microsoft security updates current to March 2013 • EC2Config service version 2.1.14 • Citrix Agent with CPU heartbeat fix • Current AWS Tools for Windows PowerShell • AWS CloudFormation helper scripts version 1.3.11

Release	Changes
2013.02.22	**All AMIs** • Microsoft security updates current to February 2013 • KB2800213 • Windows PowerShell 3.0 upgrade • EC2Config service version 2.1.13 • Citrix Agent with time fix • Citrix PV drivers dated 2011.07.19 • Current AWS Tools for Windows PowerShell • AWS CloudFormation helper scripts version 1.3.8 **Only AMIs with Microsoft SQL Server** • Microsoft SQL Server 2012 cumulative update package 5
2012.12.12	**All AMIs** • Microsoft security updates current to December 2012 • Set the ActiveTimeBias registry value to 0 • Disable IPv6 for the network adapter • EC2Config service version 2.1.9 • Add AWS Tools for Windows PowerShell and set the policy to allow import-module
2012.11.15	**All AMIs** • Microsoft security updates current to November 2012 • EC2Config service version 2.1.7
2012.10.10	**All AMIs** • Microsoft security updates current to October 2012
2012.08.15	**All AMIs** • Microsoft security updates current to August 2012 • EC2Config service version 2.1.2 • KB2545227
2012.07.11	**All AMIs** • Microsoft security updates current to July 2012

Release	Changes
2012.06.12	**All AMIs** • Microsoft security updates current to June 2012 • Set page file to 4 GB • Remove installed language packs • Set performance option to "Adjust for best performance" • Set the screen saver to no longer display the logon screen on resume • Remove previous RedHat driver versions using pnputil • Remove duplicate bootloaders and set bootstatuspolicy to ignoreallfailures using bcdedit
2012.05.10	**All AMIs** • Microsoft security updates current to May 2012 • EC2Config service version 2.1.0
2012.04.11	**All AMIs** • Microsoft security updates current to April 2012 • KB2582281 • Current version of EC2Config • System time in UTC instead of GMT
2012.03.13	**All AMIs** • Microsoft security updates current to March 2012
2012.02.24	**All AMIs** • Microsoft security updates current to February 2012 • Standardize AMI names and descriptions
2012.01.12	**All AMIs** • Microsoft security updates current to January 2012 • RedHat PV driver version 1.3.10
2011.09.11	**All AMIs** • Microsoft security updates current to September 2011

Release	Changes
1.04	**All AMIs** • Current Microsoft security updates • Update network driver • Fix issue with instances in a VPC losing connectivity when changing the time zone of the instance
1.02	**All AMIs** • Current Microsoft security updates • Update network driver • Add support for licensing activation for instances in a VPC
1.01	**All AMIs** • Current Microsoft security updates • Fix issue with password improperly generated while waiting for network availability
1.0	**All AMIs** • Initial release

Image Changes

The following changes are applied to each Amazon-provided image.

- Allow Internet Control Message Protocol (ICMP) traffic through firewall
- Set performance options for best performance
- Set power setting to high performance
- Disable screensaver password
- Disable hibernation
- Disable clearing page file at shutdown
- Add links to desktop EC2 Microsoft Windows Guide (http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/concepts.html) and EC2 Feedback (https://aws.qualtrics.com/se/?sid=sv_e5mofjhv18gtayw)
- Set timezone to UTC
- Configure page file (512 MB to 8 GB)
- Install PowerShell tools (http://aws.amazon.com/powershell)
- Install the latest version of the EC2Config service
- Disable Windows network location profile selection prompt
- Install Cloud Formation tools (http://aws.amazon.com/developertools/aws-cloudformation/4026240853893296)
- Disable IPv6 in network adapters

- Disable NetBIOS in network adapters
- Install PowerShell 3.0 for images earlier than Windows Server 2012
- Enable remote PowerShell
- Enable file and printer sharing
- Open port 1433 for images that include SQL Server
- Enable notification of Windows updates
- Sync time daily via NTP
- Disable Windows Internet Explorer RunOnce
- Apply the following hotfixes for Windows Server 2008 or Server 2008 R2 images:
  - GARP (http://support.microsoft.com/kb/2582281)
  - Microsoft DST (http://support.microsoft.com/kb/2800213)
  - Microsoft RTIU clock sync (http://support.microsoft.com/kb/2922223)
  - ELB (http://support.microsoft.com/kb/2634328)
  - TCP scaling (http://support.microsoft.com/kb/2780879)
  - SMB2 (http://support.microsoft.com/kb/2394911)
- Attach instance storage volumes to extended mount points (25)
- Install latest Microsoft Windows updates

Subscribing to Windows AMI Notifications

If you want to be notified when new AMIs are released or when the previous AMIs are made private, you can subscribe to these notifications using Amazon SNS.

To subscribe to Windows AMI notifications

1. Open the Amazon SNS console.
2. In the navigation bar, change the region to **US East (N. Virginia)**, if necessary. You must select this region because the SNS notifications that you are subscribing to were created in this region.
3. In the navigation pane, click **Subscriptions**.
4. Click **Create Subscription**.
5. In the **Create Subscription** dialog box, do the following:

   a. In **TopicARN**, enter one of the following Amazon Resource Names (ARNs):

      - arn:aws:sns:us-east-1:801119661308:ec2-windows-ami-update
      - arn:aws:sns:us-east-1:801119661308:ec2-windows-ami-private

   b. In **Protocol**, select `Email`.
   c. In **Endpoint**, enter an email address that you can use to receive the notifications.
   d. Click **Subscribe**.

6. You'll receive a confirmation email with the subject line `AWS Notification - Subscription Confirmation`. Open the email and click **Confirm subscription** to complete your subscription.

Whenever new Windows AMIs are released, we send notifications to subscribers of the `ec2-windows-ami-update` topic. Whenever new Windows AMIs are made private, we send notifications to subscribers of the `ec2-windows-ami-private` topic. If you no longer want to receive these notifications, use the following procedure to unsubscribe.

To unsubscribe from Windows AMI notifications

1. Open the Amazon SNS console.
2. In the navigation pane, click **Subscriptions**.
3. Select the subscription and then click **Delete Subscriptions** When prompted for confirmation, click **Yes, Delete**.

Upgrading or Migrating a Windows Server Instance

For information about how to upgrade or migrate an instance to a newer version of Windows, see Upgrading a Windows Server EC2 Instance to a Newer Version of Windows Server.

Create a Standard Amazon Machine Image Using Sysprep

The Microsoft System Preparation (Sysprep) tool simplifies the process of duplicating a customized installation of Microsoft Windows. We recommend that you use Sysprep to create a standardized Amazon Machine Image (AMI). You can then create new Amazon EC2 instances for Windows from this standardized image and deploy these across your organization.

We also recommend that you run Sysprep with the EC2Config service, which automates and secures the image-preparation process on your AMI by using an answer file. The file is located in the following directory, by default: C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml

Important: Do not use Sysprep to create an instance backup. Sysprep removes system-specific information; removing this information might have unintended consequences for an instance backup.

Contents

- Before You Begin (p. 90)
- Using Sysprep with the EC2Config Service (p. 90)
- Run Sysprep with the EC2Config Service (p. 94)
- Troubleshooting Sysprep with EC2Config (p. 94)

Before You Begin

- Learn more about Sysprep on Microsoft TechNet.
- Learn which server roles are supported for Sysprep.

Using Sysprep with the EC2Config Service

Learn the details of the different Sysprep execution phases and the tasks performed by the EC2Config service as the image is prepared.

Sysprep Phases

Sysprep runs through the following phases:

1. **Generalize:** The tool removes image-specific information and configurations. For example, Sysprep removes the security identifier (SID), the computer name, the event logs, and specific drivers, to name a few. After this phase is completed, the operating system (OS) is ready to create an AMI.

   Note
   When you run Sysprep with the EC2Config service, the system prevents drivers from being removed because the PersistAllDeviceInstalls setting is set to true by default.

2. **Specialize**: Plug and Play scans the computer and installs drivers for any detected devices. The tool generates OS requirements like the computer name and SID. Optionally, you can execute commands in this phase.
3. **Out-of-Box Experience (OOBE)**: The system runs an abbreviated version of Windows Setup and asks the user to enter information such as a system language, the time zone, and a registered organization. When you run Sysprep with EC2Config, the answer file automates this phase.

Sysprep Actions

Sysprep and the EC2Config service perform the following actions when preparing an image.

1. When you choose **Shutdown with Sysprep** in the **EC2 Service Properties** dialog box, the system runs the **ec2config.exe –sysprep** command.
2. The EC2Config service reads the content of the BundleConfig.xml file. This file is located in the following directory, by default: C:\Program Files\Amazon\Ec2ConfigService\Settings.

   The BundleConfig.xml file includes the following settings. You can change these settings:

   - **AutoSysprep**: Indicates whether to use Sysprep automatically. You do not need to change this value if you are running Sysprep from the EC2 Service Properties dialog box. The default value is No.
   - **SetRDPCertificate**: Sets a self-signed certificate for the Remote Desktop server running on Windows Server 2003. This enables you to securely use the Remote Desktop Protocol (RDP) to connect to the instance. Change the value to Yes if new instances should use a certificate. This setting is not used with Windows Server 2008 or Windows Server 2012 instances because these operating systems can generate their own certificates. The default value is No.
   - **SetPasswordAfterSysprep**: Sets a random password on a newly launched instance, encrypts it with the user launch key, and outputs the encrypted password to the console. Change the value to No if new instances should not be set to a random encrypted password. The default value is Yes.
   - **PreSysprepRunCmd**: The location of the command to run. The command is located in the following directory, by default: C:\Program Files\Amazon\Ec2ConfigService\Scripts\BeforeSysprep.cmd
3. The system executes the BeforeSysprep.cmd. This command creates the following registry key:

   reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 1 /f”

   The registry key disables RDP connections until they are re-enabled. Disabling RDP connections is a necessary security measure because, during the first boot session after Sysprep has run, there is a short period of time where RDP allows connections and the Administrator password is blank.
4. The EC2Config service calls sysprep.exe by executing the following command:

   sysprep.exe /unattend: "C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml" /oobe /generalize /shutdown

Generalize Phase

1. The tool removes image-specific information and configurations such as the computer name and the SID. If the instance is a member of a domain, it is removed from the domain. The sysprep2008.xml answer file includes the following settings which affect this phase:
   - **PersistAllDeviceInstalls**: This setting prevents Windows Setup from removing and reconfiguring devices, which speeds up the image preparation process because Amazon AMIs require certain drivers to run and re-detection of those drivers would take time.
   - **DoNotCleanUpNonPresentDevices**: This setting retains Plug and Play information for devices that are not currently present.
2. Sysprep.exe shuts down the OS as it prepares to create the AMI. They system either launches a new instance or starts the original instance.

Specialize Phase

The system generates OS specific requirements such as a computer name and a SID. The system also performs the following actions based on configurations that you specify in the sysprep2008.xml answer file.

- **CopyProfile**: Sysprep can be configured to delete all user profiles, including the built-in Administrator profile. This setting retains the built-in Administrator account so that any customizations you made to that account are carried over to the new image. The default value is True.

  If you don't have specific user-profile customizations that you want to carry over to the new image then change this setting to False. Sysprep will remove all user profiles; this saves time and disk space.
- **TimeZone**: The time zone is set to Coordinate Universal Time (UTC) by default.
- **Synchronous command with order 1**: The system executes the following command that enables the administrator account and specifies the password requirement.

  net user Administrator /ACTIVE:YES /LOGONPASSWORDCHG:NO /EXPIRES:NEVER /PASSWORDREQ:YES
- **Synchronous command with order 2**: The system scrambles the administrator password. This security measure is designed to prevent the instance from being accessible after Sysprep completes if you did not enable the ec2setpassword setting.

  C:\Program Files\Amazon\Ec2ConfigService\ScramblePassword.exe" -u Administrator
- **Synchronous command with order 3**: The system executes the following command:

  C:\Program Files\Amazon\Ec2ConfigService\Scripts\SysprepSpecializePhase.cmd

  This command adds the following registry key, which re-enables RDP:

  reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f

OOBE Phase

1. Using the EC2Config service answer file, the system specifies the following configurations:
   - <InputLocale>en-US</InputLocale>
   - <SystemLocale>en-US</SystemLocale>
   - <UILanguage>en-US</UILanguage>
   - <UserLocale>en-US</UserLocale>
   - <HideEULAPage>true</HideEULAPage>
   - <HideWirelessSetupInOOBE>true</HideWirelessSetupInOOBE>

- <NetworkLocation>Other</NetworkLocation>
- <ProtectYourPC>3</ProtectYourPC>
- <BluetoothTaskbarIconEnabled>false</BluetoothTaskbarIconEnabled>
- <TimeZone>UTC</TimeZone>
- <RegisteredOrganization>Amazon.com</RegisteredOrganization>
- <RegisteredOwner>Amazon</RegisteredOwner>

  Note
  During the generalize and specialize phases the EC2Config service monitors the status of of the OS. If EC2Config detects that the OS is in a Sysprep phase, then it publishes the following message the system log:
  `"EC2ConfigMonitorState: 0 Windows is being configured. SysprepState=IMAGE_STATE_UNDEPLOYABLE"`

2. After the OOBE phase completes, the system executes the SetupComplete.cmd from the following location: C:\Windows\Setup\Scripts\SetupComplete.cmd. In Amazon public AMIs before April 2015 this file was empty and executed nothing on the image. In public AMIs dated after April 2015, the file includes the following value: **call "C:\Program Files\Amazon\Ec2ConfigService\Scripts\PostSysprep.cmd"**.
3. The system executes the PostSysprep.cmd, which performs the following operations:
   - Sets the local Administrator password to not expire. If the password expired, Administrators might not be able to log on.
   - Sets the MSSQLServer machine name (if installed) so that the name will be in sync with the AMI.

Post Sysprep

After Sysprep completes, the EC2Config services sends the following message to the console output: `"Windows sysprep configuration complete. Message: Sysprep Start Message: Sysprep End"`

EC2Config then performs the following actions:

1. Reads the content of the config.xml file and lists all enabled plug-ins.
2. Executes all "Before Windows is ready" plug-ins at the same time.
   - Ec2SetPassword
   - Ec2SetComputerName
   - Ec2InitializeDrives
   - Ec2EventLog
   - Ec2ConfigureRDP
   - Ec2OutputRDPCert
   - Ec2SetDriveLetter
   - Ec2WindowsActivate
   - Ec2DynamicBootVolumeSize
3. After it is finished, sends a "Windows is ready" message to the instance system logs.
4. Runs all "After Windows is ready" plug-ins at the same time.
   - AWS CloudWatch logs
   - UserData
   - Simple Systems Manager (SSM)

For more information about Windows plug-ins, see Configuring a Windows Instance Using the EC2Config Service.

Run Sysprep with the EC2Config Service

Use the following procedure to create a standardized AMI using Sysprep and the EC2Config service.

1. In the Amazon EC2 console locate or create an AMI that you want to duplicate.
2. Launch and connect to your Windows instance.
3. Customize it.
4. Specify configuration settings in the EC2Config service answer file:

   C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml
5. From the Windows **Start** menu, choose **All Programs**, and then choose **EC2ConfigService Settings**.
6. Choose the **Image** tab in the **Ec2 Service Properties** dialog box. For more information about the options and settings in the Ec2 Service Properties dialog box, see Ec2 Service Properties.
7. Select an option for the Administrator password, and then click **Shutdown with Sysprep** or **Shutdown without Sysprep**. EC2Config edits the settings files based on the password option that you selected.
   - **Random**: EC2Config generates a password, encrypts it with user's key, and displays the encrypted password to the console. We disable this setting after the first launch so that this password persists if the instance is rebooted or stopped and started.
   - **Specify**: The password is stored in the Sysprep answer file in unencrypted form (clear text). When Sysprep runs next, it sets the Administrator password. If you shut down now, the password is set immediately. When the service starts again, the Administrator password is removed. It's important to remember this password, as you can't retrieve it later.
   - **Keep Existing**: The existing password for the Administrator account doesn't change when Sysprep is run or EC2Config is restarted. It's important to remember this password, as you can't retrieve it later.
8. Choose **OK**.

When you are asked to confirm that you want to run Sysprep and shut down the instance, click **Yes**. You'll notice that EC2Config runs Sysprep. Next, you are logged off the instance, and the instance is shut down. If you check the **Instances** page in the Amazon EC2 console, the instance state changes from `running` to `stopping`, and then finally to `stopped`. At this point, it's safe to create an AMI from this instance.

You can manually invoke the Sysprep tool from the command line using the following command:

```
C:\> %ProgramFiles%\Amazon\Ec2ConfigService\ec2config.exe -sysprep
```

However, you must be very careful that the XML file options specified in the `Ec2ConfigService\Settings` folder are correct; otherwise, you might not be able to connect to the instance. For more information about the settings files, see EC2Config Settings Files (p. 240). For an example of configuring and then running Sysprep from the command line, see `Ec2ConfigService\Scripts\InstallUpdates.ps1`.

Troubleshooting Sysprep with EC2Config

If you experience problems or receive error messages during image preparations, review the following logs:

- %WINDIR%\Panther\Unattendgc
- %WINDIR%\System32\Sysprep\Panther
- "C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2ConfigLog.txt"

If you receive an error message during image preparation with Sysprep, the OS might not be reachable. To review the log files, you must stop the instance, attach its root volume to another healthy instance as a secondary volume, and then review the logs mentioned earlier on the secondary volume.

If you locate errors in the Unattendgc log file, use the Microsoft Error Lookup Tool to get more details about the error. The following issue reported in the Unattendgc log file is typically the result of one or more corrupted user profiles on the instance:

```
Error [Shell Unattend] _FindLatestProfile failed (0x80070003) [gle=0x00000003]
Error [Shell Unattend] CopyProfile failed (0x80070003) [gle=0x00000003]
```

There are two options for resolving this issue:

Option 1: Use Regedit on the instance to search for the following key. Verify that there are no profile registry keys for a deleted user:

[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\ProfileList\

Option 2: Edit the EC2Config answer file (C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml) and change <CopyProfile>true</CopyProfile> to <CopyProfile>false</CopyProfile>. Run Sysprep again. Note that this configuration change will delete the built-in administrator user profile after Sysprep completes.

Amazon EC2 Instances

If you're new to Amazon EC2, see the following topics to get started:

- What Is Amazon EC2? (p. 1)
- Setting Up with Amazon EC2 (p. 14)
- Getting Started with Amazon EC2 Windows Instances (p. 20)
- Instance Lifecycle (p. 203)

Before you launch a production environment, you need to answer the following questions.

Q. What purchasing option best meets my needs?
Amazon EC2 supports On-Demand instances (the default), Spot instances, and Reserved Instances. For more information, see Amazon EC2 Pricing.

Q. What instance type best meets my needs?
Amazon EC2 provides different instance types to enable you to choose the CPU, memory, storage, and networking capacity that you need to run your applications. For more information, see Instance Types (p. 96).

Q. Which type of root volume meets my needs?
Each instance is backed by Amazon EBS or backed by instance store. Select an AMI based on which type of root volume you need. For more information, see Storage for the Root Device (p. 53).

Q. Would I benefit from using a virtual private cloud?
If you can launch instances in either EC2-Classic or EC2-VPC, you'll need to decide which platform meets your needs. For more information, see Supported Platforms (p. 455) and Amazon EC2 and Amazon Virtual Private Cloud (p. 449).

Instance Types

When you launch an instance, the *instance type* that you specify determines the hardware of the host computer used for your instance. Each instance type offers different compute, memory, and storage capabilities. Select an instance type based on the requirements of the application or software that you plan to run on your instance.

Amazon EC2 provides each instance with a consistent and predictable amount of CPU capacity, regardless of its underlying hardware.

Amazon EC2 dedicates some resources of the host computer, such as CPU, memory, and instance storage, to a particular instance. Amazon EC2 shares other resources of the host computer, such as the network and the disk subsystem, among instances. If each instance on a host computer tries to use as much of one of these shared resources as possible, each receives an equal share of that resource. However, when a resource is under-utilized, an instance can consume a higher share of that resource while it's available.

Each instance type provides higher or lower minimum performance from a shared resource. For example, instance types with high I/O performance have a larger allocation of shared resources. Allocating a larger share of shared resources also reduces the variance of I/O performance. For most applications, moderate I/O performance is more than enough. However, for applications that require greater or more consistent I/O performance, consider an instance type with higher I/O performance.

Contents

- Available Instance Types (p. 97)
- Hardware Specifications (p. 98)
- Networking and Storage Features (p. 98)
- Instance Limits (p. 99)

Available Instance Types

Amazon EC2 provides the instance types listed in the following tables.

Current Generation Instances

For the best performance, we recommend that you use the current generation instance types when you launch new instances. For more information about the current generation instance types, see Amazon EC2 Instances .

Instance Family	Current Generation Instance Types
General purpose	`t2.micro` | `t2.small` | `t2.medium` | `t2.large` | `m4.large` | `m4.xlarge` | `m4.2xlarge` | `m4.4xlarge` | `m4.10xlarge` | `m3.medium` | `m3.large` | `m3.xlarge` | `m3.2xlarge`
Compute optimized	`c4.large` | `c4.xlarge` | `c4.2xlarge` | `c4.4xlarge` | `c4.8xlarge` | `c3.large` | `c3.xlarge` | `c3.2xlarge` | `c3.4xlarge` | `c3.8xlarge`
Memory optimized	`r3.large` | `r3.xlarge` | `r3.2xlarge` | `r3.4xlarge` | `r3.8xlarge`
Storage optimized	`i2.xlarge` | `i2.2xlarge` | `i2.4xlarge` | `i2.8xlarge` | `d2.xlarge` | `d2.2xlarge` | `d2.4xlarge` | `d2.8xlarge`
GPU instances	`g2.2xlarge` | `g2.8xlarge`

Previous Generation Instances

Amazon Web Services offers previous generation instances for users who have optimized their applications around these instances and have yet to upgrade. We encourage you to use the latest generation of instances to get the best performance, but we will continue to support these previous generation instances. If you are currently using a previous generation instance, you can see which current generation instance would be a suitable upgrade. For more information, see Previous Generation Instances.

Instance Family	Previous Generation Instance Types
General purpose	`m1.small` | `m1.medium` | `m1.large` | `m1.xlarge`
Compute optimized	`c1.medium` | `c1.xlarge` | `cc2.8xlarge`
Memory optimized	`m2.xlarge` | `m2.2xlarge` | `m2.4xlarge` | `cr1.8xlarge`
Storage optimized	`hi1.4xlarge` | `hs1.8xlarge`
GPU instances	`cg1.4xlarge`
Micro instances	`t1.micro`

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances.

To determine which instance type best meets your needs, we recommend that you launch an instance and use your own benchmark application. Because you pay by the instance hour, it's convenient and inexpensive to test multiple instance types before making a decision.

Even after you make a decision, if your needs change, you can resize your instance later on. For more information, see Resizing Your Instance (p. 118).

Networking and Storage Features

When you select an instance type, this determines which of the following networking and storage features are available:

- Some instance types are not available in EC2-Classic, so you must launch them in a VPC. By launching an instance in a VPC, you can leverage features that are not available in EC2-Classic, such as enhanced networking, assigning multiple private IP addresses to the instance, and changing the security groups assigned to your instance. For more information, see Instance Types Available Only in a VPC (p. 455).
- Some instance types support EBS volumes and instance store volumes, while other instance types support only EBS volumes. Some instances that support instance store volumes use solid state drives (SSD) to deliver very high random I/O performance. For more information, see Storage (p. 515).
- To obtain additional, dedicated capacity for Amazon EBS I/O, you can launch some instance types as EBS–optimized instances. Some instance types are EBS–optimized by default. For more information, see Amazon EBS–Optimized Instances (p. 555).
- To optimize your instances for high performance computing (HPC) applications, you can launch some instance types in a placement group. For more information, see Placement Groups (p. 504).
- To get significantly higher packet per second (PPS) performance, lower network jitter, and lower latencies, you can enable enhanced networking for some current generation instance types. For more information, see Enabling Enhanced Networking on Windows Instances in a VPC (p. 510).
- The maximum supported MTU varies across instance types. All Amazon EC2 instance types support standard Ethernet V2 1500 MTU frames. All current generation instances support 9001 MTU, or jumbo frames, and some previous generation instances support them as well. For more information, see Network Maximum Transmission Unit (MTU) for Your EC2 Instance (p. 507).

The following table summarizes the networking and storage features supported by the current generation instance types.

	VPC only	EBS only	SSD volumes	Placement group	HVM only	Enhanced networking
C3			Yes	Yes		Yes
C4	Yes	Yes		Yes	Yes	Yes
D2				Yes	Yes	Yes
G2			Yes	Yes	Yes	
I2			Yes	Yes	Yes	Yes
M3			Yes			
M4	Yes	Yes		Yes	Yes	Yes
R3			Yes	Yes	Yes	Yes
T2	Yes	Yes			Yes	

Instance Limits

There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some instance types.

For more information about the default limits, see How many instances can I run in Amazon EC2?

For more information about viewing your current limits or requesting an increase in your current limits, see Amazon EC2 Service Limits (p. 618).

T2 Instances

T2 instances are designed to provide moderate baseline performance and the capability to burst to significantly higher performance as required by your workload. They are intended for workloads that don't use the full CPU often or consistently, but occasionally need to burst. T2 instances are well suited for general purpose workloads, such as web servers, developer environments, and small databases. For more information about T2 instance pricing and additional hardware details, see Amazon EC2 Instances.

If your account is less than 12 months old, you can use a `t2.micro` instance for free within certain usage limits. For more information, see AWS Free Tier.

Contents

- Hardware Specifications (p. 99)
- T2 Instance Requirements (p. 100)
- CPU Credits (p. 100)
- Monitoring Your CPU Credits (p. 102)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances.

T2 Instance Requirements

The following are the requirements for T2 instances:

- You must launch a T2 instance using an HVM AMI.
- You must launch your T2 instances into a virtual private cloud (VPC); they are not supported on the EC2-Classic platform. Amazon VPC enables you to launch AWS resources into a virtual network that you've defined. You cannot change the instance type of an existing instance in EC2-Classic to a T2 instance type. For more information about EC2-Classic and EC2-VPC, see Supported Platforms (p. 455) For more information about launching a VPC-only instance, see Instance Types Available Only in a VPC (p. 455).
- T2 instance types are available as Amazon EBS-backed instances only.
- T2 instances are available as On-Demand or Reserved Instances, but you can't purchase them as Spot instances. For more information, see Amazon EC2 Instance Purchasing Options.
- There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some instance types. By default, you can run up to 20 T2 instances simultaneously. If you need more T2 instances, you can request them using the Amazon EC2 Instance Request Form.

CPU Credits

A CPU Credit provides the performance of a full CPU core for one minute. Traditional Amazon EC2 instance types provide fixed performance, while T2 instances provide a baseline level of CPU performance with the ability to burst above that baseline level. The baseline performance and ability to burst are governed by CPU credits.

What is a CPU credit?

One CPU credit is equal to one vCPU running at 100% utilization for one minute. Other combinations of vCPUs, utilization, and time are also equal one CPU credit, such as one vCPU running at 50% utilization for two minutes, or two vCPUs (on `t2.medium` and `t2.large` instances, for example) running at 25% utilization for two minutes.

How are CPU credits earned?

Each T2 instance starts with a healthy initial CPU credit balance and then continuously (at a millisecond-level resolution) receives a set rate of CPU credits per hour, depending on instance size. The accounting process for whether credits are accumulated or spent also happens at a millisecond-level resolution, so you don't have to worry about overspending CPU credits; a short burst of CPU takes a small fraction of a CPU credit.

When a T2 instance uses fewer CPU resources than its base performance level allows (such as when it is idle), the unused CPU credits (or the difference between what was earned and what was spent) are stored in the credit balance for up to 24 hours, building CPU credits for bursting. When your T2 instance requires more CPU resources than its base performance level allows, it uses credits from the CPU credit balance to burst up to 100% utilization. The more credits your T2 instance has for CPU resources, the more time it can burst beyond its base performance level when more performance is needed.

The following table lists the initial CPU credit allocation received at launch, the rate at which CPU credits are received, the baseline performance level as a percentage of a full core performance, and the maximum earned CPU credit balance that an instance can accrue.

Instance type	Initial CPU credit	CPU credits earned per hour	Base performance (CPU utilization)	Maximum earned CPU credit balance***
t2.micro	30	6	10%	144
t2.small	30	12	20%	288
t2.medium	60	24	40%**	576
t2.large	60	36	60%**	864

* There are limits to how many T2 instances will launch or start with the initial CPU credit, which by default is set to 100 launches or starts of any T2 instance per account, per 24-hour period, per region. If you'd like to increase this limit, you can file a customer support limit increase request by using the Amazon EC2 Instance Request Form. If your account does not launch or start more than 100 T2 instances in 24 hours, this limit will not affect you.

** t2.medium and t2.large instances have two vCPUs. The base performance is an aggregate of the two vCPUs.

*** This maximum does not include the initial CPU credits, which are used first and do not expire. For example, a t2.micro instance that was launched and then remained idle for over 24 hours could reach a credit balance of up to 174 (30 initial CPU credits + 144 earned credits). However, once the instance uses the initial 30 CPU credits, the credit balance can never exceed 144 unless a new initial CPU credit balance is issued by stopping and starting the instance again.

The initial credit balance is designed to provide a good startup experience. The maximum earned credit balance for an instance is equal to the number of CPU credits received per hour times 24 hours. For example, a t2.micro instance earns 6 CPU credits per hour and can accumulate a maximum earned CPU credit balance of 144 CPU credits.

Do CPU credits expire?

Initial CPU credits do not expire, but they are used first when an instance uses CPU credits. Unused earned credits from a given 5 minute interval expire 24 hours after they are earned, and any expired credits are removed from the CPU credit balance at that time, before any newly earned credits are added. Additionally, the CPU credit balance for an instance does not persist between instance stops and starts; stopping an instance causes it to lose its credit balance entirely, but when it restarts it will receive its initial credit balance again.

For example, if a t2.small instance had a CPU utilization of 5% for the hour, it would have used 3 CPU credits (5% of 60 minutes), but it would have earned 12 CPU credits during the hour, so the difference of 9 CPU credits would be added to the CPU credit balance. Any CPU credits in the balance that reached their 24 hour expiration date during that time (which could be as many as 12 credits if the instance was completely idle 24 hours ago) would also be removed from the balance. If the amount of credits expired is greater than those earned, the credit balance will go down; conversely, if the amount of credits expired is fewer than those earned, the credit balance will go up.

What happens if I use all of my credits?

If your instance uses all of its CPU credit balance, performance remains at the baseline performance level. If your instance is running low on credits, your instance's CPU credit consumption (and therefore CPU performance) is gradually lowered to the base performance level over a 15-minute interval, so you will not experience a sharp performance drop-off when your CPU credits are depleted. If your instance

consistently uses all of its CPU credit balance, we recommend a larger T2 size or a fixed performance instance type such as M3 or C3.

Monitoring Your CPU Credits

You can see the credit balance for each T2 instance presented in the Amazon EC2 per-instance metrics of the CloudWatch console. T2 instances have two metrics, `CPUCreditUsage` and `CPUCreditBalance`. The `CPUCreditUsage` metric indicates the number of CPU credits used during the measurement period. The `CPUCreditBalance` metric indicates the number of unused CPU credits a T2 instance has earned. This balance is depleted during burst time as CPU credits are spent more quickly than they are earned.

The following table describes the new available CloudWatch metrics; for more information on using these metrics in CloudWatch, see View Amazon EC2 Metrics (p. 334).

Metric	Description
`CPUCreditUsage`	(Only valid for T2 instances) The number of CPU credits consumed during the specified period. This metric identifies the amount of time during which physical CPUs were used for processing instructions by virtual CPUs allocated to the instance. **Note** CPU Credit metrics are available at a 5 minute frequency. Units: Count
`CPUCreditBalance`	(Only valid for T2 instances) The number of CPU credits that an instance has accumulated. This metric is used to determine how long an instance can burst beyond its baseline performance level at a given rate. **Note** CPU Credit metrics are available at a 5 minute frequency. Units: Count

C4 Instances

C4 instances are ideal for compute-bound applications that benefit from high performance processors. C4 instances are well suited for the following applications:

- Batch processing workloads
- Media transcoding
- High-traffic web servers, massively multiplayer online (MMO) gaming servers, and ad serving engines
- High performance computing (HPC) and other compute-intensive applications

Contents

- Hardware Specifications (p. 103)
- C4 Instance Features (p. 104)
- C4 Instance Requirements (p. 104)

Hardware Specifications

C4 instances are based on custom 2.9 GHz Intel® Xeon® E5-2666 v3 (Haswell) processors, optimized specifically for Amazon EC2. With Intel® Turbo Boost Technology, the processor clock speed in C4 instances can reach as high as 3.5Ghz with 1 or 2 core Turbo Boost on `c4.8xlarge` instances.

The following table highlights the feature set of the Intel® Xeon® E5-2666 v3 processor. For more information, see Intel and Amazon Web Services.

Feature	Specification
Processor Number	E5-2666 v3
Intel® Smart Cache	25 MiB
Instruction Set	64-bit
Instruction Set Extensions	AVX 2.0
Lithography	22 nm
Processor Base Frequency	2.9 GHz
Max All Core Turbo Frequency	3.2 GHz
Max Turbo Frequency	3.5 GHz (available on `c4.8xlarge`)
Intel® Turbo Boost Technology	2.0
Intel® vPro Technology	Yes
Intel® Hyper-Threading Technology	Yes
Intel® Virtualization Technology (VT-x)	Yes
Intel® Virtualization Technology for Directed I/O (VT-d)	Yes
Intel® VT-x with Extended Page Tables (EPT)	Yes
Intel® 64	Yes
Idle States	Yes
Enhanced Intel SpeedStep® Technology	Yes
Thermal Monitoring Technologies	Yes
AES New Instructions	Yes
Secure Key	Yes
Execute Disable Bit	Yes

For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances.

C4 Instance Features

The following is a summary of the features for C4 instances:

- C4 instances are EBS-optimized by default, and deliver dedicated block storage throughput to Amazon EBS ranging from 500 Mbps to 4,000 Mbps at no additional cost. EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your C4 instance. For more information, see Amazon EBS–Optimized Instances (p. 555).
- You can enable enhanced networking capabilities. Enhanced networking provides significantly higher packet per second (PPS) performance, lower network jitter, and lower latencies. For more information, see Enabling Enhanced Networking on Windows Instances in a VPC (p. 510).
- You can cluster C4 instances in a placement group. Placement groups provide low latency and high-bandwidth connectivity between the instances within a single Availability Zone. For more information, see Placement Groups (p. 504).

C4 Instance Requirements

The following are the requirements for C4 instances:

- C4 instances require 64-bit HVM AMIs. They have high-memory (up to 60 GiB of RAM), and require a 64-bit operating system to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on high-memory instance types. In addition, you must use an HVM AMI to take advantage of enhanced networking.
- You must launch your C4 instances into a virtual private cloud (VPC); they are not supported on the EC2-Classic platform. Amazon VPC enables you to launch AWS resources into a virtual network that you've defined. For more information about EC2-Classic and EC2-VPC, see Supported Platforms (p. 455) For more information about launching a VPC-only instance, see Instance Types Available Only in a VPC (p. 455).
- There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some C4 instance types. For more information, see How many instances can I run in Amazon EC2?

  If you need more C4 instances, you can request them using the Amazon EC2 Instance Request Form.

Windows GPU Instances

If you require high parallel processing capability, you'll benefit from using GPU instances, which provide access to NVIDIA GPUs with up to 1,536 CUDA cores and 4 GB of video memory. You can use GPU instances to accelerate many scientific, engineering, and rendering applications by leveraging the Compute Unified Device Architecture (CUDA) or OpenCL parallel computing frameworks. You can also use them for graphics applications, including game streaming, 3-D application streaming, and other graphics workloads.

GPU instances run as HVM-based instances. Hardware virtual machine (HVM) virtualization uses hardware-assist technology provided by the AWS platform. With HVM virtualization, the guest VM runs as if it were on a native hardware platform, except that it still uses paravirtual (PV) network and storage drivers for improved performance. This enables Amazon EC2 to provide dedicated access to one or more discrete GPUs in each GPU instance.

You can cluster GPU instances into a placement group. Placement groups provide low latency and high-bandwidth connectivity between the instances within a single Availability Zone. For more information, see Placement Groups (p. 504).

Contents

- Hardware Specifications (p. 105)
- GPU Instance Limitations (p. 105)
- AMIs for GPU Instances (p. 105)
- Installing the NVIDIA Driver on Windows (p. 105)

For information about Linux GPU Instances, see Linux GPU Instances in the *Amazon EC2 User Guide for Linux Instances*.

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances.

GPU Instance Limitations

GPU instances have the following limitations:

- You must launch the instance using an HVM AMI.
- They can't access the GPU unless the NVIDIA drivers are installed.
- There is a limit on the number of instances that you can run. For more information, see How many instances can I run in Amazon EC2? in the Amazon EC2 FAQ. To request an increase in these limits, use the following form: Request to Increase Amazon EC2 Instance Limit.

AMIs for GPU Instances

To help you get started, NVIDIA provides AMIs for GPU instances. These reference AMIs include the NVIDIA driver, which enables full functionality and performance of the NVIDIA GPUs. For a list of AMIs with the NVIDIA driver, see AWS Marketplace (NVIDIA GRID).

You can launch CG1 and G2 instances using any HVM AMI.

Installing the NVIDIA Driver on Windows

To install the NVIDIA driver on your Windows instance, log on to your instance as the administrator using Remote Desktop. You can download NVIDIA drivers from http://www.nvidia.com/Download/Find.aspx. Select a driver for the NVIDIA GRID K520 (G2 instances) or Tesla M-Class M2050 (CG1 instances) for your version of Windows Server. Open the folder where you downloaded the driver and double-click the installation file to launch it. Follow the instructions to install the driver and reboot your instance as required. To verify that the GPU is working properly, check Device Manager.

> **Note**
> If you launch a `g2.8xlarge` instance (containing 4 GPUs) with a Windows AMI that was created on a `g2.2xlarge` instance (containing 1 GPU), Windows does not automatically install the NVIDIA driver on all 4 GPUs. You must authorize the driver installation for the new GPU hardware. You can correct this manually in the Device Manager by opening the **Other** device category (the inactive GPUs do not appear under **Display Adapters**), then for each inactive GPU, right-click and select **Update Driver Software** and choose the default **Automatic Update** option.

When using Remote Desktop, GPUs that use the WDDM driver model are replaced with a non-accelerated Remote Desktop display driver. To access your GPU hardware, you must use a different remote access

tool, such as VNC. You can also use one of the GPU AMIs from the AWS Marketplace because they provide remote access tools that support 3-D acceleration.

I2 Instances

I2 instances are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications. They are well suited for the following scenarios:

- NoSQL databases (for example, Cassandra and MongoDB)
- Clustered databases
- Online transaction processing (OLTP) systems

Contents

- Hardware Specifications (p. 106)
- I2 Instance Features (p. 106)
- I2 Instance Requirements (p. 106)
- SSD I/O Performance (p. 107)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances.

I2 Instance Features

The following is a summary of the features for I2 instances:

- The primary data storage is SSD-based instance storage. Like all instance storage, these volumes persist only for the life of the instance. When you stop or terminate an instance, the applications and data in its instance store are erased. We recommend that you regularly back up or replicate the data that you've stored in instance storage. For more information, see SSD Instance Store Volumes (p. 582).
- You can enable enhanced networking capabilities. Enhanced networking provides significantly higher packet per second (PPS) performance, lower network jitter, and lower latencies. For more information, see Enabling Enhanced Networking on Windows Instances in a VPC (p. 510).
- You can cluster I2 instances in a placement group. Placement groups provide low latency and high-bandwidth connectivity between the instances within a single Availability Zone. For more information, see Placement Groups (p. 504).
- You can enable EBS–optimization to obtain additional, dedicated capacity for Amazon EBS I/O. For more information, see Amazon EBS–Optimized Instances (p. 555).

I2 Instance Requirements

The following are the requirements for I2 instances:

- You must launch an I2 instance using an HVM AMI.
- There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some I2 instance types. For more information, see How many instances can I run in Amazon EC2?

  If you need more I2 instances, you can request them using the Amazon EC2 Instance Request Form.

SSD I/O Performance

If you use a Linux AMI with kernel version 3.8 or later and utilize all the SSD-based instance store volumes available to the instance, you can get at least the minimum random IOPS (4,096 byte block size) listed in the following table. Otherwise, you'll get lower IOPS performance than what is shown in the table.

Instance Size	Read IOPS	First Write IOPS
`i2.xlarge`	35,000	35,000
`i2.2xlarge`	75,000	75,000
`i2.4xlarge`	175,000	155,000
`i2.8xlarge`	365,000	315,000

As you fill the SSD-based instance storage for your instance, the number of write IOPS that you can achieve decreases. This is due to the extra work the SSD controller must do to find available space, rewrite existing data, and erase unused space so that it can be rewritten. This process of garbage collection results in internal write amplification to the SSD, expressed as the ratio of SSD write operations to user write operations. This decrease in performance is even larger if the write operations are not in multiples of 4,096 bytes or not aligned to a 4,096-byte boundary. If you write a smaller amount of bytes or bytes that are not aligned, the SSD controller must read the surrounding data and store the result in a new location. This pattern results in significantly increased write amplification, increased latency, and dramatically reduced I/O performance.

SSD controllers can use several strategies to reduce the impact of write amplification. One such strategy is to reserve space in the SSD instance storage so that the controller can more efficiently manage the space available for write operations. This is called *over-provisioning*. The SSD-based instance store volumes provided to an I2 instance don't have any space reserved for over-provisioning. To reduce write amplification, you should leave 10% of the volume unpartitioned so that the SSD controller can use it for over-provisioning. This decreases the storage that you can use, but increases performance.

I2 instance store–backed volumes support TRIM. You can use the TRIM command to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information, see Instance Store Volume TRIM Support (p. 582).

D2 Instances

D2 instances are designed for workloads that require high sequential read and write access to very large data sets on local storage. D2 instances are well suited for the following applications:

- Massive parallel processing (MPP) data warehouse
- MapReduce and Hadoop distributed computing
- Log or data processing applications

Contents

- Hardware Specifications (p. 108)
- D2 Instance Features (p. 108)
- D2 Instance Requirements (p. 108)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances.

D2 Instance Features

The following is a summary of the features for D2 instances:

- The primary data storage for D2 instances is HDD-based instance storage. Like all instance storage, these volumes persist only for the life of the instance. For more information about instance store volumes, see Amazon EC2 Instance Store (p. 577).
- D2 instances are EBS-optimized by default, and deliver dedicated block storage throughput to Amazon EBS ranging from 750 Mbps to 4,000 Mbps at no additional cost. EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your D2 instance. For more information, see Amazon EBS–Optimized Instances (p. 555).
- You can enable enhanced networking capabilities. Enhanced networking provides significantly higher packet per second (PPS) performance, lower network jitter, and lower latencies. For more information, see Enabling Enhanced Networking on Windows Instances in a VPC (p. 510).
- You can cluster D2 instances in a placement group. Placement groups provide low latency and high-bandwidth connectivity between the instances within a single Availability Zone. For more information, see Placement Groups (p. 504).

D2 Instance Requirements

The following are the requirements for D2 instances:

- D2 instances require 64-bit HVM AMIs. They have high-memory (up to 244 GiB of RAM), and require a 64-bit operating system to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on high-memory instance types. In addition, you must use an HVM AMI to take advantage of enhanced networking.
- There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some D2 instance types. For more information, see How many instances can I run in Amazon EC2?

  If you need more D2 instances, you can request them using the Amazon EC2 Instance Request Form.
- Your `d2.8xlarge` instances are capable of providing up to 3.5 GB/s read performance and 3.1 GB/s write performance with a 2 MiB block size.

HI1 Instances

HI1 instances (`hi1.4xlarge`) can deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications. They are well suited for the following scenarios:

- NoSQL databases (for example, Cassandra and MongoDB)
- Clustered databases
- Online transaction processing (OLTP) systems

You can cluster HI1 instances in a placement group. For more information, see Placement Groups (p. 504).

By default, you can run up to two `hi1.4xlarge` instances. If you need more than two `hi1.4xlarge` instances, you can request more using the Amazon EC2 Instance Request Form.

Contents

- Hardware Specifications (p. 109)
- Disk I/O Performance (p. 109)
- SSD Storage (p. 109)

Hardware Specifications

The `hi1.4xlarge` instance type is based on solid-state drive (SSD) technology.

For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances.

Disk I/O Performance

Using Linux paravirtual (PV) AMIs, HI1 instances can deliver more than 120,000 4 KB random read IOPS and between 10,000 and 85,000 4 KB random write IOPS (depending on active logical block addressing span) to applications across two SSD data volumes. Using hardware virtual machine (HVM) AMIs, performance is approximately 90,000 4 KB random read IOPS and between 9,000 and 75,000 4 KB random write IOPS.

HI1 Windows instances deliver approximately 90,000 4 KB random read IOPS and between 9,000 and 75,000 4 KB random write IOPS.

The maximum sequential throughput is approximately 2 GB read per second and 1.1 GB write per second.

SSD Storage

With SSD storage on HI1 instances:

- The primary data source is an instance store with SSD storage.
- Read performance is consistent and write performance can vary.
- Write amplification can occur.
- The TRIM command is not currently supported.

Instance Store with SSD Storage

The `hi1.4xlarge` instances use an Amazon EBS-backed root device. However, their primary data storage is provided by the SSD volumes in the instance store. Like other instance store volumes, these instance store volumes persist only for the life of the instance. Because the root device of the `hi1.4xlarge` instance is Amazon EBS-backed, you can still start and stop your instance. When you stop an instance, your application persists, but your production data in the instance store does not persist. For more information about instance store volumes, see Amazon EC2 Instance Store (p. 577).

Variable Write Performance

Write performance depends on how your applications utilize logical block addressing (LBA) space. If your applications use the total LBA space, write performance can degrade by about 90 percent. Benchmark your applications and monitor the queue length (the number of pending I/O requests for a volume) and I/O size.

Write Amplification

Write amplification refers to an undesirable condition associated with flash memory and SSDs, where the actual amount of physical information written is a multiple of the logical amount intended to be written. Because flash memory must be erased before it can be rewritten, the process to perform these operations results in moving (or rewriting) user data and metadata more than once. This multiplying effect increases the number of writes required over the life of the SSD, which shortens the time that it can reliably operate. The `hi1.4xlarge` instances are designed with a provisioning model intended to minimize write amplification.

Random writes have a much more severe impact on write amplification than serial writes. If you are concerned about write amplification, allocate less than the full tebibyte of storage for your application (also known as over provisioning).

The TRIM Command

The TRIM command enables the operating system to notify an SSD that blocks of previously saved data are considered no longer in use. TRIM limits the impact of write amplification.

TRIM support is not available for HI1 instances. For information about instances that support TRIM, see Instance Store Volume TRIM Support (p. 582).

HS1 Instances

HS1 instances (`hs1.8xlarge`) provide very high storage density and high sequential read and write performance per instance. They are well suited for the following scenarios:

- Data warehousing
- Hadoop/MapReduce
- Parallel file systems

You can cluster HS1 instances in a placement group. For more information, see Placement Groups (p. 504).

By default, you can run up to two HS1 instances. If you need more than two HS1 instances, you can request more using the Amazon EC2 Instance Request Form.

Contents

- Hardware Specifications (p. 110)
- Instance Store (p. 111)
- Disk Initialization (p. 111)

Hardware Specifications

HS1 instances support both Amazon Elastic Block Store (Amazon EBS)-backed and instance store-backed Amazon Machine Images (AMIs). HS1 instances support both paravirtual (PV) and hardware virtual machine (HVM) AMIs.

HS1 instances provide high bandwidth networking and can also be used with Provisioned IOPS (SSD) volumes for improved consistency and performance.

For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances.

Instance Store

HS1 instances support both instance store and Amazon EBS root device volumes. However, even when using an Amazon EBS-backed instance, primary data storage is provided by the hard disk drives in the instance store. Like other instance store volumes, these instance store volumes persist only for the life of the instance. For more information about instance store volumes, see Amazon EC2 Instance Store (p. 577).

Disk Initialization

If you plan to run an HS1 instance in a steady state for long periods of time, we recommend that you zero the hard disks first for improved performance. This process can take as long as six hours to complete.

T1 Micro Instances

T1 Micro instances (`t1.micro`) provide a small amount of consistent CPU resources and allow you to increase CPU capacity in short bursts when additional cycles are available. They are well suited for lower throughput applications and websites that require additional compute cycles periodically.

> **Note**
> The `t1.micro` is a previous generation instance and it has been replaced by the `t2.micro`, which has a much better performance profile. We recommend using the `t2.micro` instance type instead of the `t1.micro`. For more information, see T2 Instances (p. 99).

The `t1.micro` instance is available as an Amazon EBS-backed instance only.

This documentation describes how `t1.micro` instances work so that you can understand how to apply them. It's not our intent to specify exact behavior, but to give you visibility into the instance's behavior so you can understand its performance.

Topics

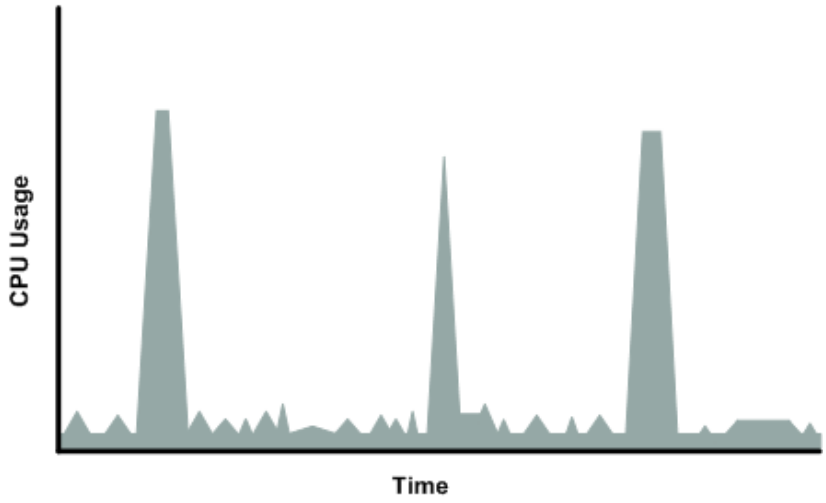
- Hardware Specifications (p. 111)
- Optimal Application of T1 Micro Instances (p. 111)
- Available CPU Resources During Spikes (p. 113)
- When the Instance Uses Its Allotted Resources (p. 114)
- Comparison with the m1.small Instance Type (p. 115)
- AMI Optimization for Micro Instances (p. 117)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances.

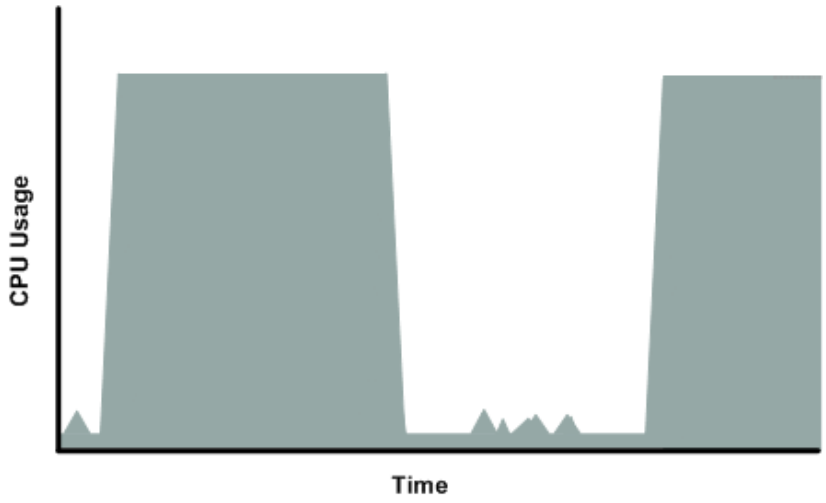
Optimal Application of T1 Micro Instances

A `t1.micro` instance provides spiky CPU resources for workloads that have a CPU usage profile similar to what is shown in the following figure.

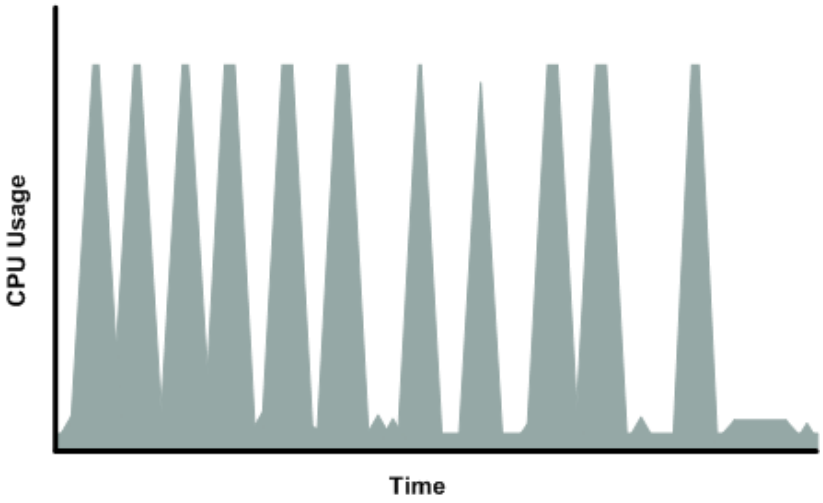


The instance is designed to operate with its CPU usage at essentially only two levels: the normal low background level, and then at brief spiked levels much higher than the background level. We allow the instance to operate at up to 2 EC2 compute units (ECUs) (one ECU provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor). The ratio between the maximum level and the background level is designed to be large. We designed `t1.micro` instances to support tens of requests per minute on your application. However, actual performance can vary significantly depending on the amount of CPU resources required for each request on your application.

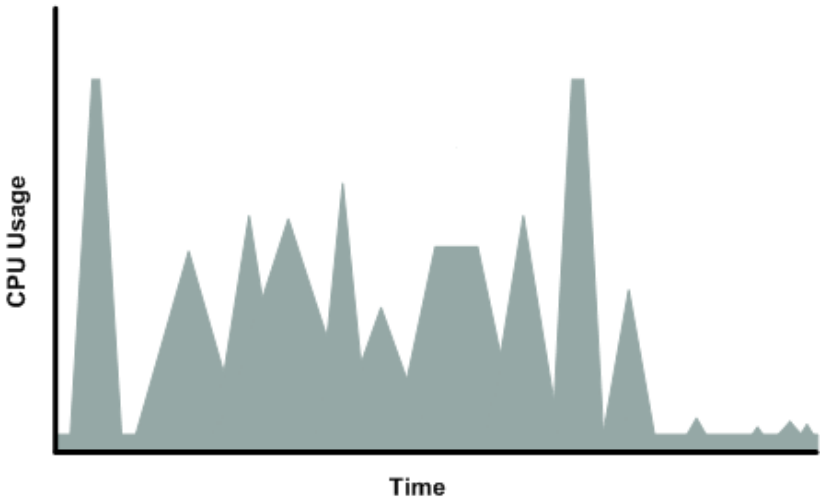
Your application might have a different CPU usage profile than that described in the preceding section. The next figure shows the profile for an application that isn't appropriate for a `t1.micro` instance. The application requires continuous data-crunching CPU resources for each request, resulting in plateaus of CPU usage that the `t1.micro` instance isn't designed to handle.



The next figure shows another profile that isn't appropriate for a `t1.micro` instance. Here the spikes in CPU use are brief, but they occur too frequently to be serviced by a micro instance.



The next figure shows another profile that isn't appropriate for a `t1.micro` instance. Here the spikes aren't too frequent, but the background level between spikes is too high to be serviced by a `t1.micro` instance.



In each of the preceding cases of workloads not appropriate for a `t1.micro` instance, we recommend that you consider using a different instance type. For more information about instance types, see Instance Types (p. 96).

Available CPU Resources During Spikes

When your instance *bursts* to accommodate a spike in demand for compute resources, it uses unused resources on the host. The amount available depends on how much contention there is when the spike occurs. The instance is never left with zero CPU resources, whether other instances on the host are spiking or not.

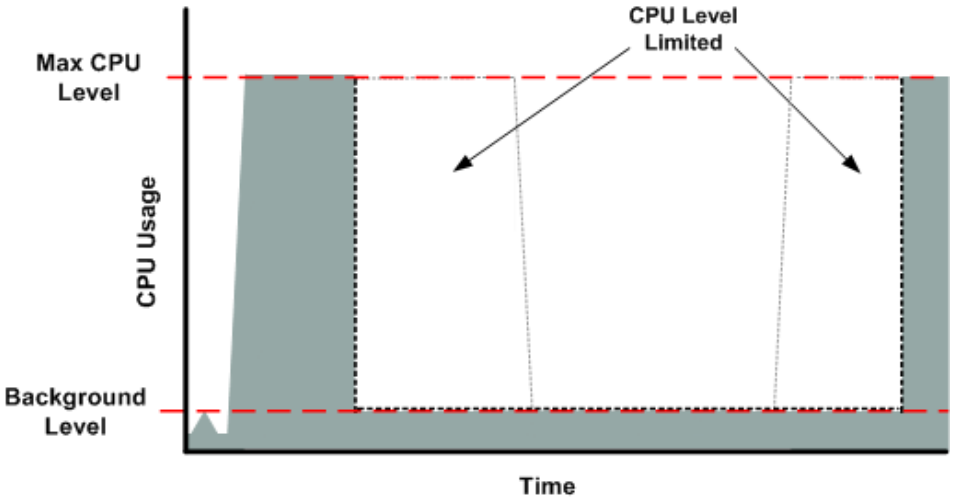
When the Instance Uses Its Allotted Resources

We expect your application to consume only a certain amount of CPU resources in a period of time. If the application consumes more than your instance's allotted CPU resources, we temporarily limit the instance so it operates at a low CPU level. If your instance continues to use all of its allotted resources, its performance will degrade. We will increase the time that we limit its CPU level, thus increasing the time before the instance is allowed to burst again.

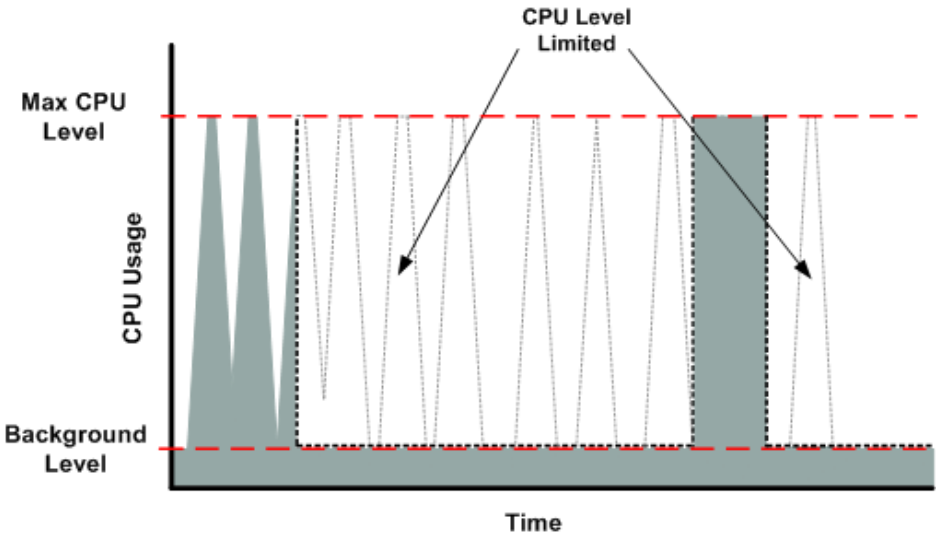
If you enable CloudWatch monitoring for your `t1.micro` instance, you can use the "Avg CPU Utilization" graph in the AWS Management Console to determine whether your instance is regularly using all its allotted CPU resources. We recommend that you look at the maximum value reached during each given period. If the maximum value is 100%, we recommend that you use Auto Scaling to scale out (with additional `t1.micro` instances and a load balancer), or move to a larger instance type. For more information, see the Auto Scaling Developer Guide.

The following figures show the three suboptimal profiles from the preceding section and what it might look like when the instance consumes its allotted resources and we have to limit its CPU level. If the instance consumes its allotted resources, we restrict it to the low background level.

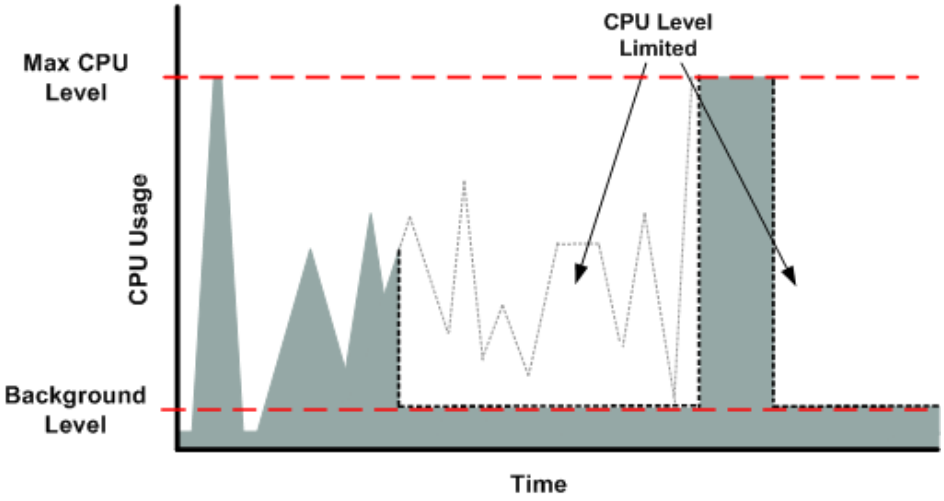
The next figure shows the situation with the long plateaus of data-crunching CPU usage. The CPU hits the maximum allowed level and stays there until the instance's allotted resources are consumed for the period. At that point, we limit the instance to operate at the low background level, and it operates there until we allow it to burst above that level again. The instance again stays there until the allotted resources are consumed and we limit it again (not seen on the graph).



The next figure shows the situation where the requests are too frequent. The instance uses its allotted resources after only a few requests and so we limit it. After we lift the restriction, the instance maxes out its CPU usage trying to keep up with the requests, and we limit it again.

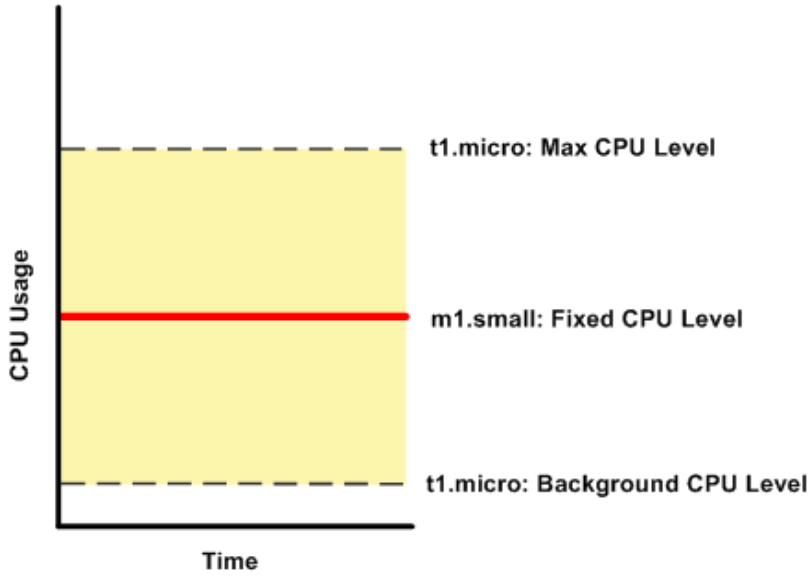


The next figure shows the situation where the background level is too high. Notice that the instance doesn't have to be operating at the maximum CPU level for us to limit it. We limit the instance when it's operating above the normal background level and has consumed its allotted resources for the given period. In this case (as in the preceding one), the instance can't keep up with the work, and we limit it again.



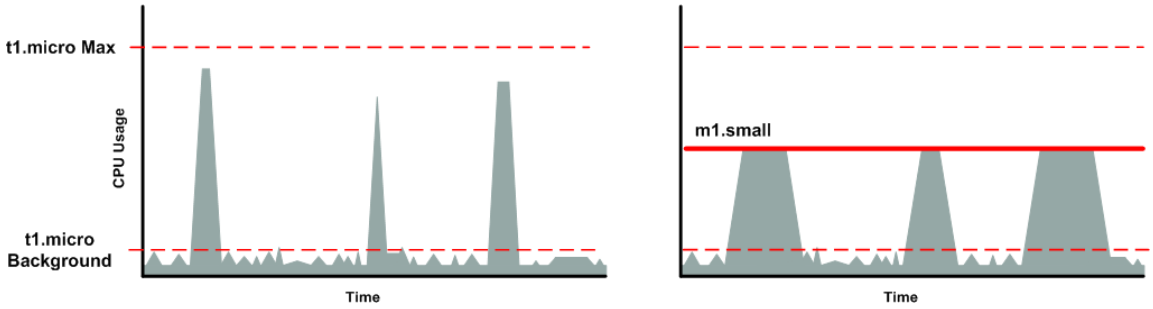
Comparison with the m1.small Instance Type

The `t1.micro` instance provides different levels of CPU resources at different times (up to 2 ECUs). By comparison, the `m1.small` instance type provides 1 ECU at all times. The following figure illustrates the difference.

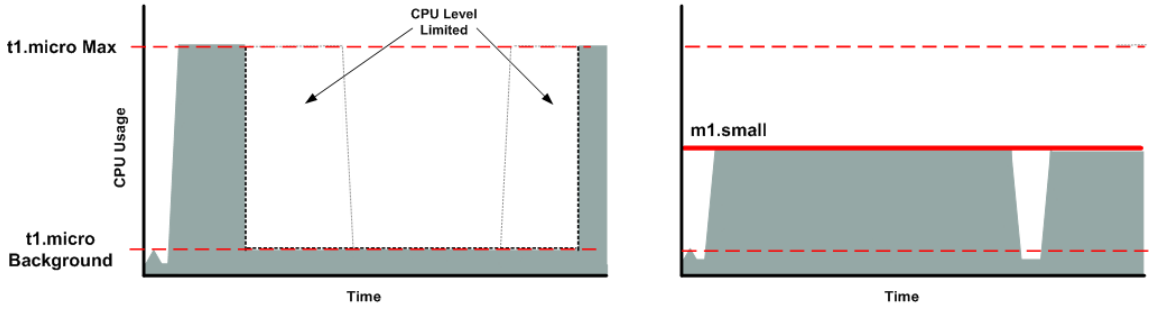


The following figures compare the CPU usage of a `t1.micro` instance with an `m1.small` instance for the various scenarios we've discussed in the preceding sections.

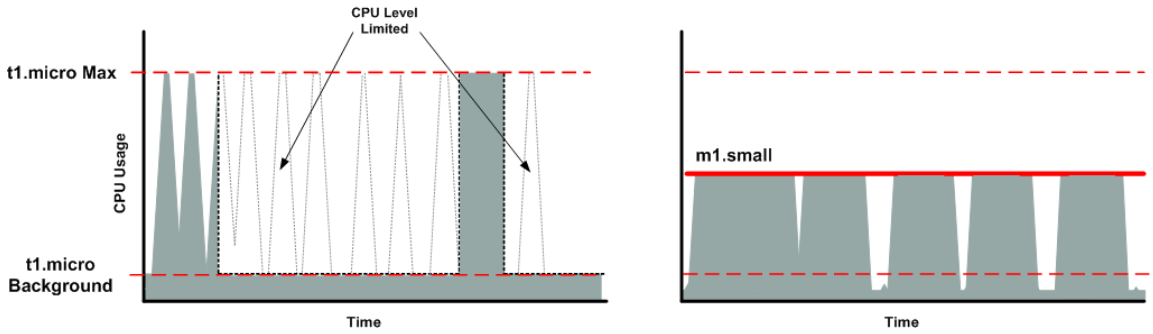
The first figure that follows shows an optimal scenario for a `t1.micro` instance (the left graph) and how it might look for an `m1.small` instance (the right graph). In this case, we don't need to limit the `t1.micro` instance. The processing time on the `m1.small` instance would be longer for each spike in CPU demand compared to the `t1.micro` instance.



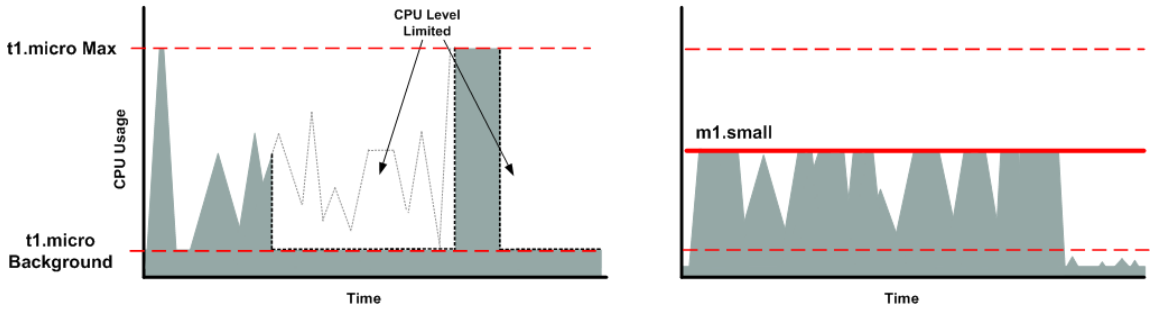
The next figure shows the scenario with the data-crunching requests that used up the allotted resources on the `t1.micro` instance, and how they might look with the `m1.small` instance.



The next figure shows the frequent requests that used up the allotted resources on the `t1.micro` instance, and how they might look on the `m1.small` instance.



The next figure shows the situation where the background level used up the allotted resources on the `t1.micro` instance, and how it might look on the `m1.small` instance.



AMI Optimization for Micro Instances

We recommend that you follow these best practices when optimizing an AMI for the `t1.micro` instance type:

- Design the AMI to run on 600 MB of RAM
- Limit the number of recurring processes that use CPU time (for example, cron jobs, daemons)

When you perform significant AMI or instance configuration changes (for example, enable server roles or install large applications), you might see limited instance performance, because these changes can be memory intensive and require long-running CPU resources. We recommend that you first use a larger

instance type when performing these changes to the AMI, and then run the AMI on a `t1.micro` instance for normal operations.

Resizing Your Instance

As your needs change, you might find that your instance is over-utilized (the instance type is too small) or under-utilized (the instance type is too large). If this is the case, you can change the size of your instance. For example, if your `t2.micro` instance is too small for its workload, you can change it to an `m3.medium` instance.

If the root device for your instance is an EBS volume, you can change the size of the instance simply by changing its instance type, which is known as *resizing* it. If the root device for your instance is an instance store volume, you must migrate your application to a new instance with the instance type that you want. For more information about root device volumes, see Storage for the Root Device (p. 53).

When you resize an instance, you must select an instance type that is compatible with the configuration of the instance. If the instance type that you want is not compatible with the instance configuration you have, then you must migrate your application to a new instance with the instance type that you want.

> **Important**
> When you resize an instance, you can't add instance store volumes; the resized instance has the same instance store volumes that you specified when you launched it. If you want to add instance store volumes, you must migrate your application to a new instance with the instance type and instance store volumes that you want. For more information about instance store volumes, see Amazon EC2 Instance Store (p. 577).

Contents

- Compatibility for Resizing Instances (p. 118)
- Resizing an Amazon EBS–backed Instance (p. 118)
- Migrating an Instance Store-backed Instance (p. 120)
- Migrating to a New Instance Configuration (p. 121)

Compatibility for Resizing Instances

You can resize an instance only if its current instance type and the new instance type that you want are compatible in the following ways:

- Network. Some instance types are not supported in EC2-Classic and must be launched in a VPC. Therefore, you can't resize an instance in EC2-Classic to a instance type that is available only in a VPC unless you have a nondefault VPC. For more information, see Instance Types Available Only in a VPC (p. 455).
- Platform. All Amazon EC2 instance types support 64-bit AMIs, but only the following instance types support 32-bit AMIs: `t2.micro`, `t2.small`, `t2.medium`, `c3.large`, `t1.micro`, `m1.small`, `m1.medium`, and `c1.medium`. If you are resizing a 32-bit instance, you are limited to these instance types.

For example, T2 instances are not supported in EC2-Classic and they are HVM only. Therefore, you can't resize a T1 instance to a T2 instance because T1 instances do not support HVM and must be launched from PV AMIs. If you want to resize a T2 instance to a larger instance type, you can select any current generation instance type, such as M3, because all current generation instance types support HVM AMIs. For more information, see Available Instance Types (p. 97).

Resizing an Amazon EBS–backed Instance

You must stop your Amazon EBS–backed instance before you can change its instance type. When you stop and start an instance, be aware of the following:

- We move the instance to new hardware; however, the instance ID does not change.
- If your instance is running in a VPC and has a public IP address, we release the address and give it a new public IP address. The instance retains its private IP addresses and any Elastic IP addresses.
- If your instance is running in EC2-Classic, we give it new public and private IP addresses, and disassociate any Elastic IP address that's associated with the instance. Therefore, to ensure that your users can continue to use the applications that you're hosting on your instance uninterrupted, you must re-associate any Elastic IP address after you restart your instance.
- If your instance is in an Auto Scaling group, the Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance. To prevent this, you can suspend the Auto Scaling processes for the group while you're resizing your instance. For more information, see Suspend and Resume Auto Scaling Processes in the *Auto Scaling Developer Guide*.

For more information, see Stop and Start Your Instance (p. 218).

Use the following procedure to resize an Amazon EBS–backed instance using the AWS Management Console.

To resize an Amazon EBS–backed instance

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances**, and select the instance.
3. [EC2-Classic] If the instance has an associated Elastic IP address, write down the Elastic IP address and the instance ID shown in the details pane.
4. Choose **Actions**, select **Instance State**, and then choose **Stop**.
5. In the confirmation dialog box, choose **Yes, Stop**. It can take a few minutes for the instance to stop.

   [EC2-Classic] When the instance state becomes `stopped`, the **Elastic IP**, **Public DNS**, **Private DNS**, and **Private IPs** fields in the details pane are blank to indicate that the old values are no longer associated with the instance.
6. With the instance still selected, choose **Actions**, select **Instance Settings**, and then choose **Change Instance Type**. Note that this action is disabled if the instance state is not `stopped`.
7. In the **Change Instance Type** dialog box, do the following:

   a. From **Instance Type**, select the instance type that you want. If the instance type that you want does not appear in the list, then it is not compatible with the configuration of your instance (for example, because of virtualization type).
   b. (Optional) If the instance type that you selected supports EBS–optimization, select **EBS-optimized** to enable EBS–optimization or deselect **EBS-optimized** to disable EBS–optimization. Note that if the instance type that you selected is EBS–optimized by default, **EBS-optimized** is selected and you can't deselect it.
   c. Choose **Apply** to accept the new settings.

8. To restart the stopped instance, select the instance, choose **Actions**, select **Instance State**, and then choose **Start**.
9. In the confirmation dialog box, choose **Yes, Start**. It can take a few minutes for the instance to enter the `running` state.
10. [EC2-Classic] When the instance state is `running`, the **Public DNS**, **Private DNS**, and **Private IPs** fields in the details pane contain the new values that we assigned to the instance. If your instance had an associated Elastic IP address, you must reassociate it as follows:

    a. In the navigation pane, choose **Elastic IPs**.
    b. Select the Elastic IP address that you wrote down before you stopped the instance.
    c. Choose **Actions** and then choose **Associate Address**.

d. From **Instance**, select the instance ID that you wrote down before you stopped the instance, and then choose **Associate**.

Migrating an Instance Store-backed Instance

When you want to move your application from one instance store-backed instance to an instance store-backed instance with a different instance type, you must migrate it by creating an image from your instance, and then launching a new instance from this image with the instance type that you need. To ensure that your users can continue to use the applications that you're hosting on your instance uninterrupted, you must take any Elastic IP address that you've associated with your original instance and associate it with the new instance. Then you can terminate the original instance.

To migrate an instance store-backed instance

1. [EC2-Classic] If the instance you are migrating has an associated Elastic IP address, record the Elastic IP address now so that you can associate it with the new instance later.
2. Back up any data on your instance store volumes that you need to keep to persistent storage. To migrate data on your EBS volumes that you need to keep, take a snapshot of the volumes (see Creating an Amazon EBS Snapshot (p. 550)) or detach the volume from the instance so that you can attach it to the new instance later (see Detaching an Amazon EBS Volume from an Instance (p. 542)).
3. Create an AMI from your instance store-backed instance by satisfying the prerequisites and following the procedures in Creating an Instance Store-Backed Windows AMI (p. 70). When you are finished creating an AMI from your instance, return to this procedure.
4. Open the Amazon EC2 console and in the navigation pane, select **AMIs**. From the filter lists, select **Owned by me**, and select the image that you created in the previous step. Notice that **AMI Name** is the name that you specified when you registered the image and **Source** is your Amazon S3 bucket.

   Note
   If you do not see the AMI that you created in the previous step, make sure that you have selected the region in which you created your AMI.
5. Choose **Launch**. When you specify options for the instance, be sure to select the new instance type that you want. If the instance type that you want can't be selected, then it is not compatible with configuration of the AMI that you created (for example, because of virtualization type). You can also specify any EBS volumes that you detached from the original instance.

   Note that it can take a few minutes for the instance to enter the `running` state.
6. [EC2-Classic] If the instance that you started with had an associated Elastic IP address, you must associate it with the new instance as follows:

   a. In the navigation pane, choose **Elastic IPs**.
   b. Select the Elastic IP address that you recorded at the beginning of this procedure.
   c. Choose **Actions** and then choose **Associate Address**.
   d. From **Instance**, select the new instance, and then choose **Associate**.
7. (Optional) You can terminate the instance that you started with, if it's no longer needed. Select the instance and verify that you are about to terminate the original instance, not the new instance (for example, check the name or launch time). Choose **Actions**, select **Instance State**, and then choose **Terminate**.

Migrating to a New Instance Configuration

If the current configuration of your instance is incompatible with the new instance type that you want, then you can't resize the instance to that instance type. Instead, you can migrate your application to a new instance with a configuration that is compatible with the new instance type that you want.

If you want to move from an instance launched from a PV AMI to an instance type that is HVM only, the general process is as follows:

1. Back up any data on your instance store volumes that you need to keep to persistent storage. To migrate data on your EBS volumes that you need to keep, create a snapshot of the volumes (see Creating an Amazon EBS Snapshot (p. 550)) or detach the volume from the instance so that you can attach it to the new instance later (see Detaching an Amazon EBS Volume from an Instance (p. 542)).
2. Launch a new instance, selecting the following:
   - An HVM AMI.
   - The HVM only instance type.
   - [EC2-VPC] If you are using an Elastic IP address, select the VPC that the original instance is currently running in.
   - Any EBS volumes that you detached from the original instance and want to attach to the new instance, or new EBS volumes based on the snapshots that you created.
   - If you want to allow the same traffic to reach the new instance, select the security group that is associated with the original instance.
3. Install your application and any required software on the instance.
4. Restore any data that you backed up from the instance store volumes of the original instance.
5. If you are using an Elastic IP address, assign it to the newly launched instance as follows:
   a. In the navigation pane, choose **Elastic IPs**.
   b. Select the Elastic IP address that is associated with the original instance, choose **Actions**, and then choose **Disassociate Address**. When prompted for confirmation, choose **Yes, Disassociate**.
   c. With the Elastic IP address still selected, choose **Actions**, and then choose **Associate Address**.
   d. From **Instance**, select the new instance, and then choose **Associate**.
6. (Optional) You can terminate the original instance if it's no longer needed. Select the instance and verify that you are about to terminate the original instance, not the new instance (for example, check the name or launch time). Choose **Actions**, select **Instance State**, and then choose **Terminate**.

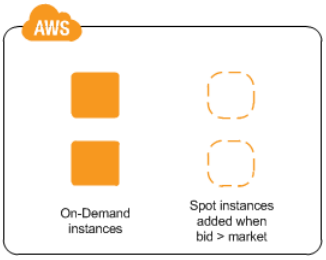
For information about migrating an application from an instance in EC2-Classic to an instance in a VPC, see Migrating from a Windows Instance in EC2-Classic to a Windows Instance in a VPC (p. 465).

Spot Instances

Spot instances enable you to bid on unused EC2 instances, which can lower your Amazon EC2 costs significantly. The hourly price for a Spot instance (of each instance type in each Availability Zone) is set by Amazon EC2, and fluctuates depending on the supply of and demand for Spot instances. Your Spot instance runs whenever your bid exceeds the current market price.

Spot instances are a cost-effective choice if you can be flexible about when your applications run and if your applications can be interrupted. For example, Spot instances are well-suited for data analysis, batch jobs, background processing, and optional tasks. For more information, see Amazon EC2 Spot Instances.

The key differences between Spot instances and On-Demand instances are that Spot instances might not start immediately, the hourly price for Spot instances varies based on demand, and Amazon EC2 can terminate an individual Spot instance as the hourly price for or availability of Spot instances changes. One strategy is to launch a core group of On-Demand instances to maintain a minimum level of guaranteed compute resources for your applications, and supplement them with Spot instances when the opportunity arises.



Concepts

Before you get started with Spot instances, you should be familiar with the following concepts:

- *Spot pool*—A set of unused EC2 instances with the same instance type, operating system, Availability Zone, and network platform (EC2-Classic or EC2-VPC).
- *Spot price*—The current market price of a Spot instance per hour, which is set by Amazon EC2 based on the last fulfilled bid. You can also retrieve the Spot price history.
- *Spot instance request* (or *Spot bid*)—Provides the maximum price (bid price) that you are willing to pay per hour for a Spot instance. When your bid price exceeds the Spot price, Amazon EC2 fulfills your request. Note that a Spot instance request is either *one-time* or *persistent*. Amazon EC2 automatically resubmits a persistent Spot request after the Spot instance associated with the request is terminated.
- *Spot fleet*—A set of Spot instances that is launched based on criteria that you specify. The Spot fleet selects the Spot pools that meet your needs and launches Spot instances to meet the target capacity for the fleet. The Spot fleet also maintains the target capacity of the fleet over time by launching replacement instances after Spot instances in the fleet are terminated.
- *Spot instance interruption*—Amazon EC2 terminates your Spot instance when the Spot price exceeds your bid price or there are no longer any unused EC2 instances. You can't enable termination protection for Spot instances.
- *Bid status*—Provides detailed information about the current state of your Spot bid.

How to Get Started

The first thing you need to do is get set up to use Amazon EC2. It can also be helpful to have experience launching On-Demand instances before launching Spot instances.

Get Up and Running

- Setting Up with Amazon EC2 (p. 14)
- Getting Started with Amazon EC2 Windows Instances (p. 20)

Spot Basics

- How Spot Instances Work (p. 124)
- How Spot Fleet Works (p. 127)

Working with Spot Instances

- Preparing for Interruptions (p. 155)
- Creating a Spot Instance Request (p. 134)
- Getting Bid Status Information (p. 153)

Working with Spot Fleets

- Spot Fleet Prerequisites (p. 139)
- Creating a Spot Fleet Request (p. 141)

Related Services

You can provision Spot instances directly using Amazon EC2. You can also provision Spot instances using other services in AWS. For more information, see the following documentation.

Auto Scaling and Spot instances
: You can create launch configurations with a bid price so that Auto Scaling can launch Spot instances. For more information, see Launching Spot instances in Your Auto Scaling Group in the *Auto Scaling Developer Guide*.

Amazon EMR and Spot instances
: There are scenarios where it can be useful to run Spot instances in an Amazon EMR cluster. For more information, see Lower Costs with Spot Instances in the *Amazon Elastic MapReduce Developer Guide*.

AWS CloudFormation Templates
: AWS CloudFormation enables you to create and manage a collection of AWS resources using a template in JSON format. AWS CloudFormation templates can include a Spot price. For more information, see EC2 Spot Instance Updates - Auto Scaling and CloudFormation Integration.

AWS SDK for Java
: You can use the Java programming language to manage your Spot instances. For more information, see Tutorial: Amazon EC2 Spot Instances and Tutorial: Advanced Amazon EC2 Spot Request Management.

AWS SDK for .NET
: You can use the .NET programming environment to manage your Spot instances. For more information, see Tutorial: Amazon EC2 Spot instances.

Pricing

You pay the Spot price for Spot instances, which is set by Amazon EC2 and fluctuates periodically depending on the supply of and demand for Spot instances. If your bid price exceeds the current Spot price, Amazon EC2 fulfills your request and your Spot instances run until either you terminate them or the Spot price increases above your bid price.

Everyone pays that same Spot price for that period, regardless of whether their bid price was higher. You never pay more than your bid price per hour, and often pay less per hour. For example, if you bid $0.25 per hour, and the Spot price is $0.20 per hour, you only pay $0.20 per hour. If the Spot price drops, you pay the new, lower price. If the Spot price rises, you pay the new price if it is equal to or less than your bid price. If the Spot price rises above your bid price, then your Spot instance is interrupted.

At the start of each instance hour, you are billed based on the Spot price. If your Spot instance is interrupted in the middle of an instance hour because the Spot price exceeded your bid, you are not billed for the partial hour of use. If you terminate your Spot instance in the middle of an instance hour, you are billed for the partial hour of use.

To view the current (updated every five minutes) lowest Spot price per region and instance type, see the Spot Instances page.

To view the Spot price history for the past three months, use the Amazon EC2 console or the describe-spot-price-history command (AWS CLI). For more information, see Spot Instance Pricing History (p. 132).

To review your bill, go to your AWS Account Activity page. Your bill contains links to usage reports that provide details about your bill. For more information, see AWS Account Billing.

If you have questions concerning AWS billing, accounts, and events, contact AWS Support.

How Spot Instances Work

To use Spot instances, create a *Spot instance request* or a *Spot fleet request*), which includes the maximum price that you are willing to pay per hour per instance (your bid price), and other constraints such as the instance type and Availability Zone. If your bid price is greater than the current Spot price for the specified instance, and the specified instance is available, your request is fulfilled immediately. Otherwise, the request is fulfilled whenever the Spot price falls below your bid price or the specified instance becomes available. Spot instances run until you terminate them or until Amazon EC2 must terminate them (also known as a *Spot instance interruption*).

When you use Spot instances, you must be prepared for interruptions. Amazon EC2 can interrupt your Spot instance when the Spot price rises above your bid price, when the demand for Spot instances rises, or when the supply of Spot instances decreases. For more information, see Spot Instance Interruptions (p. 155).

Note that you can't stop and start an Amazon EBS-backed instance if it is a Spot instance, but you can reboot or terminate it.

Contents

- Supply and Demand in the Spot Market (p. 125)
- Launching Spot instances in a Launch Group (p. 126)
- Launching Spot Instances in an Availability Zone Group (p. 126)
- Launching Spot Instances in a VPC (p. 127)

Supply and Demand in the Spot Market

AWS continuously evaluates how many Spot instances are available in each Spot pool, monitors the bids that have been made for each Spot pool, and provisions the available Spot instances to the highest bidders. The Spot price for a Spot pool is set to the lowest fulfilled bid for that pool. Therefore, the Spot price is the price above which you must bid to fulfill a Spot request for a single Spot instance immediately.

For example, suppose that you create a Spot instance request, and that the corresponding Spot pool has only five Spot instances for sale. Your bid price is $0.10, which is also the current Spot price. The following table shows the current bids, ranked in descending order. Bids 1-5 are fulfilled. Bid 5, being the last fulfilled bid, sets the Spot price at $0.10. Bid 6 is unfulfilled. Bids 3-5, which share the same bid price of $0.10, are ranked in random order.

Bid	Bid price	Current Spot price	Notes
1	$1.00	$0.10	
2	$1.00	$0.10	
3	$0.10	$0.10	
4	$0.10	$0.10	Your bid
5	$0.10	$0.10	Last fulfilled bid, which sets the Spot price. Everyone pays the same Spot price for the period.
— — —	— — —		Spot capacity cutoff
6	$0.05		

Now, let's say that the size of this Spot pool drops to 3. Bids 1-3 are fulfilled. Bid 3, the last fulfilled bid, sets the Spot price at $0.10. Bids 4-5, which also are $0.10, are unfulfilled. As you can see, even though the Spot price didn't change, two of the bids, including your bid, are no longer fulfilled because the Spot supply decreased.

Bid	Bid price	Current Spot price	Notes
1	$1.00	$0.10	
2	$1.00	$0.10	
3	$0.10	$0.10	Last fulfilled bid, which sets the Spot price. Everyone pays the same Spot price for the period.
— — —	— — —		Spot capacity cutoff
4	$0.10		Your bid
5	$0.10		
6	$0.05		

To fulfill a Spot request for a single instance from this pool, you must bid above the current Spot price of $0.10. If you bid $0.101, your request will be fulfilled, the Spot instance for bid 3 would be interrupted,

and the Spot price would become $0.101. If you bid $2.00, the Spot instance for bid 3 would be interrupted and the Spot price would become $1.00 (the price for bid 2).

Keep in mind that no matter how high you bid, you can never get more than the available number of Spot instances in the Spot pool. If the size of the pool drops to zero, then all the Spot instances from that pool would be interrupted.

Launching Spot instances in a Launch Group

Specify a launch group in your Spot instance request to tell Amazon EC2 to launch a set of Spot instances only if it can launch them all. In addition, if the Spot service must terminate one of the instances in a launch group (for example, if the Spot price rises above your bid price), it must terminate them all. However, if you terminate one or more of the instances in a launch group, Amazon EC2 does not terminate the remaining instances in the launch group.

Note that although this option can be useful, adding this constraint can lower the chances that your Spot instance request is fulfilled. It can also increase the chance that your Spot instances will be terminated.

If you create another successful Spot instance request that specifies the same (existing) launch group as an earlier successful request, then the new instances are added to the launch group. Subsequently, if an instance in this launch group is terminated, all instances in the launch group are terminated, which includes instances launched by the first and second requests.

Launching Spot Instances in an Availability Zone Group

Specify an Availability Zone group in your Spot instance request to tell the Spot service to launch a set of Spot instances in the same Availability Zone. Note that Amazon EC2 need not terminate all instances in an Availability Zone group at the same time. If Amazon EC2 must terminate one of the instances in an Availability Zone group, the others remain running.

Note that although this option can be useful, adding this constraint can lower the chances that your Spot instance request is fulfilled.

If you specify an Availability Zone group but don't specify an Availability Zone in the Spot instance request, the result depends on whether you specified the EC2-Classic network, a default VPC, or a nondefault VPC. For more information about EC2-Classic and EC2-VPC, see Supported Platforms (p. 455).

EC2-Classic

Amazon EC2 finds the lowest-priced Availability Zone in the region and launches your Spot instances in that Availability Zone if the lowest bid for the group is higher than the current Spot price in that Availability Zone. Amazon EC2 waits until there is enough capacity to launch your Spot instances together, as long as the Spot price remains lower than the lowest bid for the group.

Default VPC

Amazon EC2 uses the Availability Zone for the specified subnet, or if you don't specify a subnet, it selects an Availability Zone and its default subnet, but it might not be the lowest-priced Availability Zone. If you deleted the default subnet for an Availability Zone, then you must specify a different subnet.

Nondefault VPC

Amazon EC2 uses the Availability Zone for the specified subnet.

Launching Spot Instances in a VPC

To take advantage of the features of EC2-VPC when you use Spot instances, specify in your Spot request that your Spot instances are to be launched in a VPC. You specify a subnet for your Spot instances the same way that you specify a subnet for your On-Demand instances.

The process for making a Spot instance request that launches Spot instances in a VPC is the same as the process for making a Spot instance request that launches Spot instances in EC2-Classic—except for the following differences:

- You should base your bid on the Spot price history of Spot instances in a VPC.
- [Default VPC] If you want your Spot instance launched in a specific low-priced Availability Zone, you must specify the corresponding subnet in your Spot instance request. If you do not specify a subnet, Amazon EC2 selects one for you, and the Availability Zone for this subnet might not have the lowest Spot price.
- [Nondefault VPC] You must specify the subnet for your Spot instance.

How Spot Fleet Works

A *Spot fleet* is a collection, or fleet, of Spot instances. The Spot fleet attempts to launch the number of Spot instances that are required to meet the target capacity that you specified in the Spot fleet request. The Spot fleet also attempts to maintain its target capacity fleet if your Spot instances are interrupted due to a change in Spot prices or available capacity.

A *Spot pool* is a set of unused EC2 instances with the same instance type, operating system, Availability Zone, and network platform (EC2-Classic or EC2-VPC). When you make a Spot fleet request, you can include multiple launch specifications, that vary by instance type, AMI, Availability Zone, or subnet. The Spot fleet selects the Spot pools that are used to fulfill the request, based on the launch specifications included in your Spot fleet request, and the configuration of the Spot fleet request. The Spot instances come from the selected Spot pools.

Contents

- Spot Fleet Allocation Strategy (p. 127)
- Spot Price Overrides (p. 128)
- Spot Fleet Instance Weighting (p. 128)
- Walkthrough: Using Spot Fleet with Instance Weighting (p. 129)

Spot Fleet Allocation Strategy

The allocation strategy for your Spot fleet determines how it fulfills your Spot fleet request from the possible Spot pools represented by its launch specifications. The following are the allocation strategies that you can specify in your Spot fleet request:

`lowestPrice`
: The Spot instances come from the Spot pool with the lowest price. This is the default strategy.

`diversified`
: The Spot instances are distributed across all Spot pools.

Choosing an Allocation Strategy

You can optimize your Spot fleets based on your use case.

If your fleet is small or runs for a short time, the probability that your Spot instances will be interrupted is low, even with all the instances in a single Spot pool. Therefore, the `lowestPrice` strategy is likely to meet your needs while providing the lowest cost.

If your fleet is large or runs for a long time, you can improve the availability of your fleet by distributing the Spot instances across multiple Spot pools. For example, if your Spot fleet request specifies 10 pools and a target capacity of 100 instances, the Spot fleet launches 10 Spot instances in each pool. If the Spot price for one pool increases above your bid price for this pool, only 10% of your fleet is affected. Using this strategy also makes your fleet less sensitive to increases in the Spot price in any one pool over time.

Note that with the `diversified` strategy, the Spot fleet does not launch Spot instances into any Spot pools with a Spot price that is higher than the On-Demand price.

Maintaining Target Capacity

When Spot instances are terminated due to a change in the Spot price or available capacity of a Spot pool, the Spot fleet launches replacement Spot instances. If the allocation strategy is `lowestPrice`, the Spot fleet launches replacement instances in the pool where the Spot price is currently the lowest. If the allocation strategy is `diversified`, the Spot fleet distributes the replacement Spot instances across the remaining pools.

Spot Price Overrides

Each Spot fleet request must include a global Spot price. By default, the Spot fleet uses this price as the bid price for each of its launch specifications.

You can optionally specify a Spot price in one or more launch specifications. This bid price is specific to the launch specification. If a launch specification includes a specific Spot price, the Spot fleet uses this price as the bid price for that launch specification, overriding the global Spot price. Note that any other launch specifications that do not include a specific Spot price still use the global Spot price.

Spot Fleet Instance Weighting

When you request a fleet of Spot instances, you can define the capacity units that each instance type would contribute to your application's performance, and adjust your bid price for each Spot pool accordingly using *instance weighting*.

By default, the Spot price that you specify represents your bid price *per instance hour*. When you use the instance weighting feature, the Spot price that you specify represents your bid price *per unit hour*. You can calculate your bid price per unit hour by dividing your bid price for an instance type by the number of units that it represents. The Spot fleet calculates the number of Spot instances to launch by dividing the target capacity by the instance weight. If the result isn't an integer, the Spot fleet rounds it up to the next integer, so that the size of your fleet is not below its target capacity.

The following table includes examples of calculations to determine the bid price per unit for a Spot fleet request with a target capacity of 10.

Instance type	Instance weight	Spot price per instance hour	Spot price per unit hour	Number of instances launched
r3.xlarge	2	$0.05	.025 (.05 divided by 2)	5 (10 divided by 2)

Instance type	Instance weight	Spot price per instance hour	Spot price per unit hour	Number of instances launched
r3.8xlarge	8	$0.10	.0125 (.10 divided by 8)	2 (10 divided by 8, result rounded up)

Use Spot fleet instance weighting as follows to provision the target capacity you want in the Spot pools with the lowest price per unit at the time of fulfillment:

1. Set the target capacity for your Spot fleet either in instances (the default) or in the units of your choice, such as virtual CPUs, memory, storage, or throughput.
2. Set the bid price per unit.
3. For each launch configuration, specify the weight, which is the number of units that the instance type represents toward the target capacity.

Instance Weighting Example

Consider a Spot fleet request with the following configuration:

- A target capacity of 24
- A launch specification with an instance type `r3.2xlarge` and a weight of 6
- A launch specification with an instance type `c3.xlarge` and a weight of 5

The weights represent the number of units that instance type represents toward the target capacity. If the first launch specification provides the lowest Spot price per unit (Spot price for `r3.2xlarge` per instance hour divided by 6), the Spot fleet would launch four of these instances (24 divided by 6).

If the second launch specification provides the lowest Spot price per unit (Spot price for `c3.xlarge` per instance hour divided by 5), the Spot fleet would launch five of these instances (24 divided by 5, result rounded up).

Instance Weighting and Allocation Strategy

Consider a Spot fleet request with the following configuration:

- A target capacity of 30
- A launch specification with an instance type `c3.2xlarge` and a weight of 8
- A launch specification with an instance type `m3.xlarge` and a weight of 8
- A launch specification with an instance type `r3.xlarge` and a weight of 8

The Spot fleet would launch four instances (30 divided by 8, result rounded up). With the `lowestPrice` strategy, all four instances come from the Spot pool that provides the lowest Spot price per unit. With the `diversified` strategy, the Spot fleet launches 1 instance in each of the three pools, and the fourth instance in whichever of the three pools provides the lowest Spot price per unit.

Walkthrough: Using Spot Fleet with Instance Weighting

This walkthrough uses a fictitious company called Example Corp to illustrate the process of bidding for a Spot fleet using instance weighting.

Objective

Example Corp, a pharmaceutical company, wants to leverage the computational power of Amazon EC2 for screening chemical compounds that might be used to fight cancer.

Planning

Example Corp first reviews Spot Best Practices. Next, Example Corp determines the following requirements for their Spot fleet.

Instance Types

Example Corp has a compute- and memory-intensive application that performs best with at least 60 GB of memory and eight virtual CPUs (vCPUs). They want to maximize these resources for the application at the lowest possible price. Example Corp decides that any of the following EC2 instance types would meet their needs:

Instance type	Memory (GiB)	vCPUs
r3.2xlarge	61	8
r3.4xlarge	122	16
r3.8xlarge	244	32

Target Capacity in Units

With instance weighting, target capacity can equal a number of instances (the default) or a combination of factors such as cores (vCPUs), memory (GiBs), and storage (GBs). By considering the base for their application (60 GB of RAM and eight vCPUs) as 1 unit, Example Corp decides that 20 times this amount would meet their needs. So the company sets the target capacity of their Spot fleet request to 20.

Instance Weights

After determining the target capacity, Example Corp calculates instance weights. To calculate the instance weight for each instance type, they determine the units of each instance type that are required to reach the target capacity as follows:

- r3.2xlarge (61.0 GB, 8 vCPUs) = 1 unit of 20
- r3.4xlarge (122.0 GB, 16 vCPUs) = 2 units of 20
- r3.8xlarge (244.0 GB, 32 vCPUs) = 4 units of 20

Therefore, Example Corp assigns instance weights of 1, 2, and 4 to the respective launch configurations in their Spot fleet request.

Bid Price Per Unit Hour

Example Corp uses the On-Demand price per instance hour as a starting point for their bid price. They could also use recent Spot prices, or a combination of the two. To calculate bid price per unit hour, they divide their starting bid price per instance hour by the weight. For example:

Instance type	On-Demand price	Instance weight	Price per unit hour
r3.2xLarge	$0.7	1	$0.7
r3.4xLarge	$1.4	2	$0.7

Instance type	On-Demand price	Instance weight	Price per unit hour
r3.8xLarge	$2.8	4	$0.7

Example Corp could enter a global bid price per unit hour of $0.7 and be competitive for all three instance types. They could also enter a global bid price per unit hour of $0.7 and a specific bid price per unit hour of $0.9 in the `r3.8xlarge` launch specification. Depending on the strategy for provisioning their Spot fleet, Example Corp could bid lower to further reduce costs, or bid higher to reduce the probability of interruption.

Verifying Permissions

Before creating a Spot fleet request, Example Corp verifies that it has an IAM role with the required permissions. For more information, see Spot Fleet Prerequisites (p. 139).

Creating the Request

Example Corp creates a file, `config.json`, with the following configuration for its Spot fleet request:

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.2xlarge",
      "SubnetId": "subnet-482e4972",
      "WeightedCapacity": 1
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.4xlarge",
      "SubnetId": "subnet-482e4972",
      "WeightedCapacity": 2
    }
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.8xlarge",
      "SubnetId": "subnet-482e4972",
      "SpotPrice": "0.90",
      "WeightedCapacity": 4
    }
  ]
}
```

Example Corp creates the Spot fleet request using the following request-spot-fleet command:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

For more information, see Spot Fleet Requests (p. 138).

Fulfillment

The allocation strategy determines which Spot pools your Spot instances come from.

With the `lowestPrice` strategy (which is the default strategy), the Spot instances come from the Spot pool with the lowest Spot price per unit at the time of fulfillment. To provide 20 units of capacity, the Spot fleet launches either 20 `r3.2xlarge` instances (20 divided by 1), 10 `r3.4xlarge` instances (20 divided by 2), or 5 `r3.8xlarge` instances (20 divided by 4).

If Example Corp used the `diversified` strategy, the Spot instances would come from all three Spot pools. The Spot fleet would launch 6 `r3.2xlarge` instances (which provide 6 units), 3 `r3.4xlarge` instances (which provide 6 units), and 2 `r3.8xlarge` instances (which provide 8 units), for a total of 20 units.

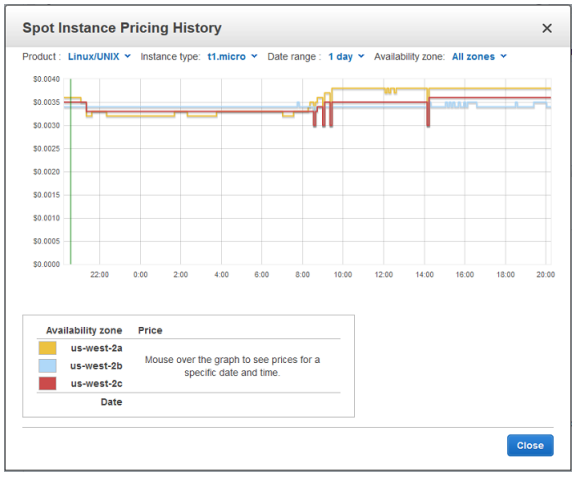
Spot Instance Pricing History

The Spot price represents the price above which you have to bid to guarantee that a single Spot request is fulfilled. When your bid price is above the Spot price, Amazon EC2 launches your Spot instance, and when the Spot price rises above your bid price, Amazon EC2 terminates your Spot instance. You can bid above the current Spot price so that your Spot request is fulfilled quickly. However, before you specify a bid price for your Spot instance, we recommend that you review the Spot price history. You can view the Spot price history for the last 90 days, filtering by instance type, operating system, and Availability Zone.

Using the Spot price history as a guide, you can select a bid price that would have met your needs in the past. For example, you can determine which bid price that would have provided 75 percent uptime in the time range you viewed. However, keep in mind that the historical trends are not a guarantee of future results. Spot prices vary based on real-time supply and demand, and the conditions that generated certain patterns in the Spot price might not occur in the future.

To view the Spot price history using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Spot Requests**.
3. Click **Pricing History**. By default, the page displays a graph of the data for Linux `t1.micro` instances in all Availability Zones over the past day. Move your mouse over the graph to display the prices at specific times in the table below the graph.



4. (Optional) To review the Spot price history for a specific Availability Zone, select an Availability Zone from the list. You can also select a different product, instance type, or date range.

To view the Spot price history using the command line

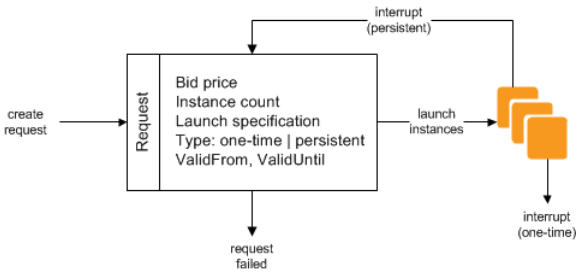
You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-spot-price-history (AWS CLI)
- ec2-describe-spot-price-history (Amazon EC2 CLI)
- Get-EC2SpotPriceHistory (AWS Tools for Windows PowerShell)

Spot Instance Requests

To use Spot instances, you create a Spot instance request that includes the number of instances, the instance type, the Availability Zone, and the maximum price that you are willing to pay per instance hour (your bid). If your bid exceeds the current Spot price, Amazon EC2 fulfills your request immediately. Otherwise, Amazon EC2 waits until your request can be fulfilled or until you cancel the request.

The following illustration shows how Spot requests work. Notice that the action taken for a Spot instance interruption depends on the request type (one-time or persistent). If the request is a persistent request, the request is opened again after your Spot instance is terminated.



Contents

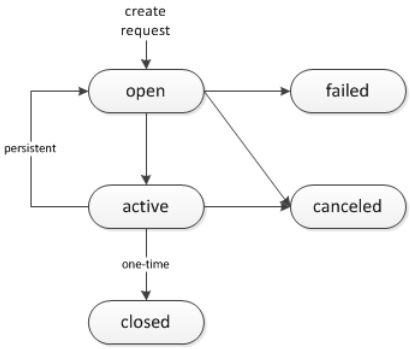
- Spot Instance Request States (p. 133)
- Creating a Spot Instance Request (p. 134)
- Finding Running Spot Instances (p. 136)
- Tagging Spot Instance Requests (p. 136)
- Canceling a Spot Instance Request (p. 137)

Spot Instance Request States

A Spot instance request can be in one of the following states:

- `open`—The request is waiting to be fulfilled.
- `active`—The request is fulfilled and has an associated Spot instance.
- `failed`—The request has one or more bad parameters.
- `closed`—The Spot instance was interrupted or terminated.
- `canceled`—You canceled the request, or the request expired.

The following illustration represents the transitions between the request states. Notice that the transitions depend on the request type (one-time or persistent).



A one-time Spot instance request remains active until Amazon EC2 launches the Spot instance, the request expires, or you cancel the request. If the Spot price rises above your bid price, your Spot instance is terminated and the Spot instance request is closed.

A persistent Spot instance request remains active until it expires or you cancel it, even if the request is fulfilled. For example, if you create a persistent Spot instance request for one instance when the Spot price is $0.25, Amazon EC2 launches your Spot instance if your bid price is above $0.25. If the Spot price rises above your bid price, your Spot instance is terminated; however, the Spot instance request is open again and Amazon EC2 launches a new Spot instance when the Spot price falls below your bid price.

You can track the status of your Spot instance requests, as well as the status of the Spot instances launched, through the bid status. For more information, see Spot Bid Status (p. 150).

Creating a Spot Instance Request

The process for requesting a Spot instance is similar to the process for launching an On-Demand instance. Note that you can't change the parameters of your Spot request, including the bid price, after you've submitted the request.

If you request multiple Spot instances at one time, Amazon EC2 creates separate Spot instance requests so that you can track the status of each request separately. For more information about tracking Spot requests, see Spot Bid Status (p. 150).

Prerequisites

Before you begin, decide on your bid price, how many Spot instances you'd like, and what instance type to use.

To create a Spot instance request using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Spot Requests**.
3. (Optional) To review Spot price trends, click **Pricing History**. For more information, see Spot Instance Pricing History (p. 132).
4. Click **Request Spot instances**.
5. On the **Choose an Amazon Machine Image** page, select an AMI.
6. On the **Choose an Instance Type** page, select an instance type and then click **Next: Configure Instance Details**.
7. On the **Configure Instance Details** page, do the following:

   a. (Optional) By default, the request launches one Spot instance. To launch multiple Spot instances, specify the number of Spot instances to launch.

   b. The purchasing option **Request Spot Instances** is select by default. To create a Spot request, leave this option selected.

c. In **Maximum price**, specify the price that you are willing to pay for your Spot instance. If your bid price exceeds the Spot price, the Spot request launches the Spot instance immediately.

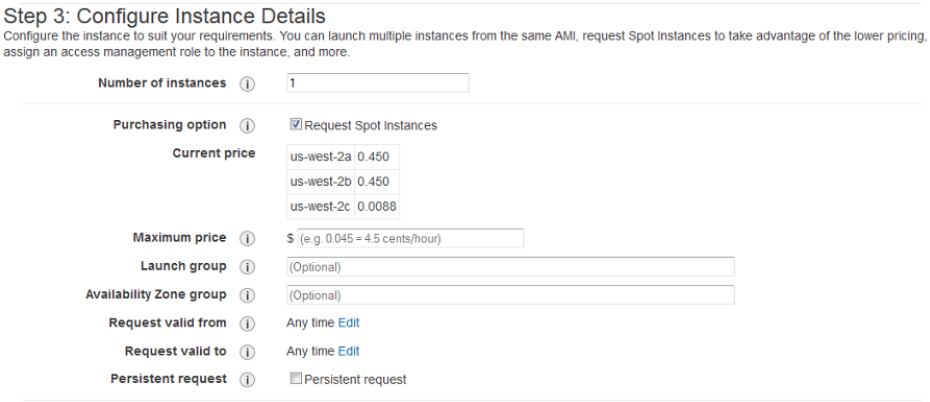
   Notice that the current Spot price for each Availability Zones in the region is listed for your information.

d. (Optional) By default, the request remains in effect until it is fulfilled or you cancel it. To create a request that is valid only during a specific time period, specify **Request valid from** and **Request valid to**.

e. (Optional) By default, the request is a one-time request. To create a persistent Spot request, select **Persistent request**.

f. (Optional) If you specified multiple instances, you can also specify a launch group or an Availability Zone group.

g. Specify the remaining options as you would for an On-Demand instance, and then click **Review and Launch**. If you are prompted to specify the type of root volume, make your selection and then click **Next**.



8. On the **Review Instance Launch** page, click **Edit security groups**. On the **Configure Security Group** page, click **Select an existing security group**, select or create a security group, and then click **Review and Launch**.
9. On the **Review Instance Launch** page, click **Launch**.
10. In the **Select an existing key pair or create a new key pair** dialog box, select **Choose an existing key pair**, then select or create a key pair. Click the acknowledgment check box, and then click **Request Spot Instances**.
11. On the confirmation page, click **View Spot Requests**. In the **Description** tab, notice that the request state is `open` and the request status is `pending-evaluation` to start. After the request is fulfilled, the request state is `active` and the request status is `fulfilled`.

To create a Spot instance request using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- request-spot-instances (AWS CLI)
- ec2-request-spot-instances (Amazon EC2 CLI)
- Request-EC2SpotInstance (AWS Tools for Windows PowerShell)

Finding Running Spot Instances

Amazon EC2 launches a Spot instance when the Spot price is below your bid. A Spot instance runs until either its bid price is no longer higher than the Spot price, or you terminate it yourself. (If your bid price is exactly equal to the Spot price, there is a chance that your Spot instance will remain running, depending on demand.)

To find running Spot instances using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Spot Requests** and select the request. If the request has been fulfilled, the value of the **Instance** column is the ID of the Spot instance.
3. Alternatively, in the navigation pane, click **Instances**. In the top right corner, click the **Show/Hide** icon, and then select **Lifecycle**. The value of the **Lifecycle** column for each instance is either `normal` or `spot`.

To find running Spot instances using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-spot-instance-requests (AWS CLI)
- describe-instances with --filters "Name=instance-lifecycle,Values=spot" (AWS CLI)
- ec2-describe-spot-instance-requests (Amazon EC2 CLI)
- ec2-describe-instances with --filter "instance-lifecycle=spot" (Amazon EC2 CLI)

Tagging Spot Instance Requests

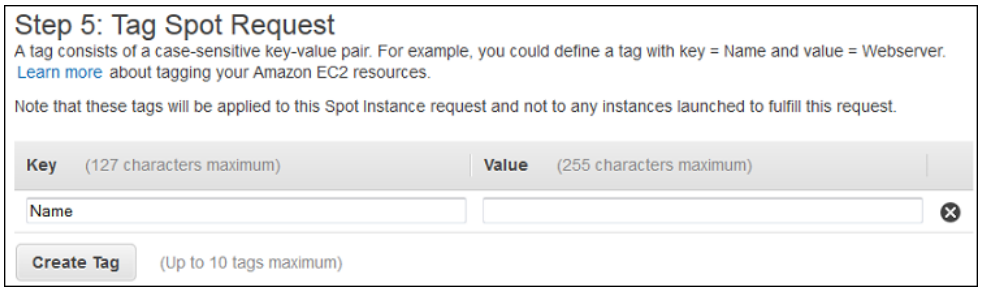
To help categorize and manage your Spot instance requests, you can tag them with metadata of your choice. You tag your Spot instance requests in the same way that you tag other any other Amazon EC2 resource. For more information, see Tagging Your Amazon EC2 Resources (p. 609).

You can tag your Spot instance request when you first create it, or you can assign a tag to the request after you create it.

The tags that you create for your Spot instance requests only apply to the requests. These tags are not added automatically to the Spot instance that the Spot service launches to fulfill the request. You must add tags to a Spot instance yourself when you create the Spot instance request or after the Spot instance is launched.

To add a tag when creating a Spot instance request

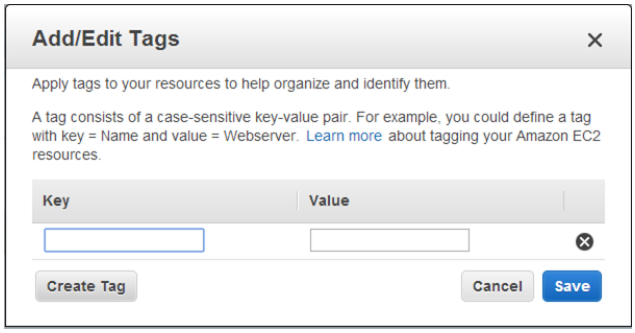
1. When creating a Spot instance request using the console, on the **Review** page, click **Edit tags**. You can also complete the tag with the key `Name` by adding a name for the Spot instance request as the value.
2. On the **Tag Spot Request** page, click **Create Tag** and enter a tag key and tag value.



3. Click **Review and Launch**.

To add a tag to an existing Spot instance request using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Spot Requests** and then select the Spot request.
3. From the **Tags** tab, click **Add/Edit Tags**.
4. In the **Add/Edit Tags** dialog box, click **Create Tag**, specify the key and value for each tag, and then click **Save**.



5. (Optional) You can add tags to the Spot instance launched from the Spot request. On the **Spot Requests** page, click the ID of the Spot instance in the **Instance** column for the Spot request. In the bottom pane, select the **Tags** tab and repeat the process that you used to add tags to the Spot request.

To create a tag for your Spot instance request or Spot instances using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- create-tags (AWS CLI)
- ec2-create-tags (Amazon EC2 CLI)

Canceling a Spot Instance Request

If you no longer want your Spot request, you can cancel it. You can only cancel Spot instance requests that are `open` or `active`. Your Spot request is `open` when your request has not yet been fulfilled and no instances have been launched. Your Spot request is `active` when your request has been fulfilled, and Spot instances have launched as a result. If your Spot request is `active` and has an associated running Spot instance, canceling the request does not terminate the instance; you must terminate the running Spot instance manually.

If the Spot request is a persistent Spot request, it returns to the open state so that a new Spot instance can be launched. To cancel a persistent Spot request and terminate its Spot instances, you must cancel the Spot request first and then terminate the Spot instances. Otherwise, the Spot request can launch a new instance.

To cancel a Spot instance request using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Spot Requests**, and then select the Spot request.
3. Click **Cancel**.
4. (Optional) If you are finished with the associated Spot instances, you can terminate them. In the navigation pane, click **Instances**, select the instance, click **Actions**, select **Instance State**, and then click **Terminate**.

To cancel a Spot instance request using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- cancel-spot-instance-requests (AWS CLI)
- ec2-cancel-spot-instance-requests (Amazon EC2 CLI)
- Stop-EC2SpotInstanceRequest (AWS Tools for Windows PowerShell)

Spot Fleet Requests

To use a Spot fleet, you create a Spot fleet request that includes the target capacity, one or more launch specifications for the instances, and the bid price that you are willing to pay. Your bid price is per instance hour (by default) or per unit hour if your request uses Spot Fleet Instance Weighting (p. 128). Amazon EC2 attempts to maintain your Spot fleet's target capacity as Spot prices change. For more information, see How Spot Fleet Works (p. 127).

Each launch specification includes the information that Amazon EC2 needs to launch an instance—such as an AMI, an instance type, a subnet or Availability Zone, and one or more security groups.

A Spot fleet request remains active until it expires or you cancel it. By default, canceling a Spot fleet request doesn't affect the Spot instances in your Spot fleet. Alternatively, you can specify that canceling your Spot fleet request terminates the Spot instances in your Spot fleet.

Contents

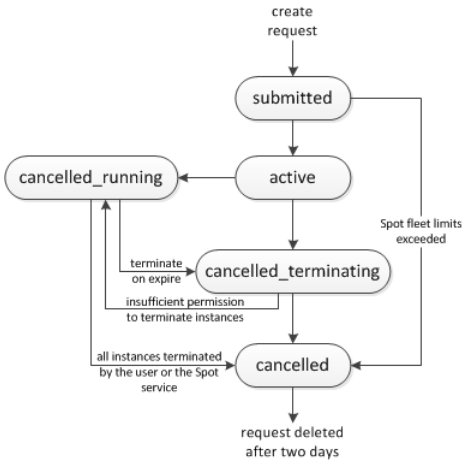
- Spot Fleet Request States (p. 138)
- Spot Fleet Prerequisites (p. 139)
- Planning a Spot Fleet Request (p. 140)
- Creating a Spot Fleet Request (p. 141)
- Monitoring Your Spot Fleet (p. 141)
- Canceling a Spot Fleet Request (p. 141)
- Spot Fleet Example Configurations (p. 142)

Spot Fleet Request States

A Spot fleet request can be in one of the following states:

- `submitted`—The Spot fleet request is being evaluated and Amazon EC2 is preparing to launch the target number of Spot instances.
- `active`—The Spot fleet has been validated and Amazon EC2 is attempting to maintain the target number of running Spot instances.
- `cancelled-running`—The Spot fleet is canceled and will not launch additional Spot instances, but its existing Spot instances will continue to run until they are interrupted or terminated. This is the default behavior when canceling a Spot fleet.
- `cancelled-terminating`—The Spot fleet is canceled and its Spot instances are terminating. This is an option when canceling a Spot fleet.
- `cancelled`—The Spot fleet is canceled and has no running Spot instances. The Spot fleet request will be deleted two days after its instances were terminated.

The following illustration represents the transitions between the request states.



Spot Fleet Prerequisites

Before you begin, complete the following:

1. Create an IAM role that grants the Spot fleet permission to bid on, launch, and terminate instances on your behalf. You specify this role, by Amazon Resource Name (ARN), when you create a Spot fleet request.

   a. Open the IAM console at https://console.aws.amazon.com/iam/.
   b. In the navigation pane, choose **Roles**, and then choose **Create New Role**.
   c. On the **Set Role Name** page, enter a name for the role and then choose **Next Step**.
   d. On the **Select Role Type** page, choose **Select** next to **Amazon EC2 Spot Fleet Role**.
   e. On the **Attach Policy** page, select the `AmazonEC2SpotFleetRole` policy, and then choose **Next Step**.
   f. On the **Review** page, choose **Create Role**.

2. Confirm that the user account that you will use to create the Spot fleet request has the required Amazon EC2 permissions:

   a. Open the IAM console at https://console.aws.amazon.com/iam/.
   b. In the navigation pane, choose **Policies**, and then choose **Create Policy**.
   c. On the **Create Policy** page, choose **Select** next to **Create Your Own Policy**.

d. On the **Review Policy** page, enter a policy name and copy the following text into the **Policy Document** section.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*"
        }
    ]
}
```

The `iam:PassRole` action enables an IAM user to pass the Spot fleet role you created above when submitting a Spot fleet request. If you want to restrict this user to launch only Spot instances, you can choose to explicitly list the following actions:

```
"Action": [
 "ec2:CancelSpotInstanceRequests",
 "ec2:CreateTags",
 "ec2:RequestSpotInstances",
 "ec2:RunInstances",
 "ec2:TerminateInstances"
]
```

e. Choose **Create Policy**.

f. In the navigation pane, choose **Users**, and then choose the user who will submit the Spot fleet request.

g. On the **User** page, in the **Permissions** section, choose **Attach Policy**.

h. On the **Attach Policy** page, select the policy you created above, and choose **Attach Policy**.

3. Install and configure an SDK or a command line interface, as the Amazon EC2 console does not support Spot fleet. For more information, see Accessing Amazon EC2 (p. 3).

Planning a Spot Fleet Request

Before you create a Spot fleet request, review Spot Best Practices. Use these best practices when you plan your Spot fleet request so that you can provision the type of instances you want at the lowest possible price. We also recommend that you do the following:

- Determine the instance types that meet your application requirements.
- Determine the target capacity for your Spot fleet request. You can set target capacity in instances or in custom units. For more information, see Spot Fleet Instance Weighting (p. 128).

- Determine your bid price per instance hour. Bidding at or near the On-Demand price is a good starting point. Bidding lower can further reduce costs, while bidding higher can reduce the probability of interruption.
- Determine your bid price per unit, if you are using instance weighting. To calculate the bid price per unit, divide the bid price per instance hour by the number of units (or weight) that this instance represents. (If you are not using instance weighting, the default bid price per unit is the bid price per instance hour.)
- Review the possible options for your Spot fleet request. For more information, see the request-spot-fleet command in the *AWS Command Line Interface Reference*. For additional examples, see Spot Fleet Example Configurations (p. 142).

Creating a Spot Fleet Request

Use the request-spot-fleet command to create a Spot fleet request:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

The following is example output:

```
{
    "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

Monitoring Your Spot Fleet

Use the following describe-spot-fleet-requests command to describe your Spot fleet requests:

```
aws ec2 describe-spot-fleet-requests
```

Use the following describe-spot-fleet-instances command to describe the Spot instances for the specified Spot fleet:

```
aws ec2 describe-spot-fleet-instances --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Use the following describe-spot-fleet-request-history command to describe the history for the specified Spot fleet request:

```
aws ec2 describe-spot-fleet-request-history --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --start-time 2015-05-18T00:00:00Z
```

Canceling a Spot Fleet Request

When you are finished using your Spot fleet resources, you can cancel the Spot fleet request. This cancels all Spot requests associated with the Spot fleet, so no new Spot instances are launched for your Spot fleet. By default, the existing Spot instances in your Spot fleet continue to run until they are interrupted or terminated. Alternatively, when you cancel the Spot fleet request, you can specify the option to terminate all Spot instances for the Spot fleet.

Example: Cancel the request and terminate the Spot instances

Use the following `cancel-spot-fleet-requests` command to cancel the specified Spot fleet request and terminate the instances:

```
aws ec2 cancel-spot-fleet-requests --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --terminate-instances
```

The following is example output:

```
{
    "SuccessfulFleetRequests": [
        {
            "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
            "CurrentSpotFleetRequestState": "cancelled_terminating",
            "PreviousSpotFleetRequestState": "active"
        }
    ],
    "UnsuccessfulFleetRequests": []
}
```

Example: Cancel the request but keep the Spot instances

Use the following cancel-spot-fleet-requests command to cancel the specified Spot fleet request without terminating the instances:

```
aws ec2 cancel-spot-fleet-requests --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --no-terminate-instances
```

The following is example output:

```
{
    "SuccessfulFleetRequests": [
        {
            "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
            "CurrentSpotFleetRequestState": "cancelled_running",
            "PreviousSpotFleetRequestState": "active"
        }
    ],
    "UnsuccessfulFleetRequests": []
}
```

Spot Fleet Example Configurations

The following examples show launch configurations that you can use with request-spot-fleet command to create a Spot fleet request. For more information, see Creating a Spot Fleet Request (p. 141).

1. Launch Spot instances using the lowest-priced Availability Zone or subnet in the region (p. 143)
2. Launch Spot instances using the lowest-priced Availability Zone or subnet in a specified list (p. 143)
3. Launch Spot instances using the lowest-priced instance type in a specified list (p. 145)
4. Override the Spot price for the request (p. 146)
5. Launch a Spot fleet using the diversified allocation strategy (p. 147)
6. Launch a Spot fleet using instance weighting (p. 148)

Example 1: Launch Spot instances using the lowest-priced Availability Zone or subnet in the region

The following example specifies a single launch configuration without an Availability Zone or subnet. If your account supports EC2-VPC only, the Spot fleet launches the instances in the lowest-priced Availability Zone that has a default subnet. If your account supports EC2-Classic, the Spot fleet launches the instances in EC2-Classic in the lowest-priced Availability Zone. Note that the price you pay will not exceed the specified Spot price for the request.

```
{
    "SpotPrice": "0.07",
    "TargetCapacity": 20,
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "SecurityGroups": [
                {
                    "GroupId": "sg-0a42d66a"
                }
            ],
            "InstanceType": "m3.medium",
            "IamInstanceProfile": {
               "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"

            }
        }
    ]
}
```

Example 2: Launch Spot instances using the lowest-priced Availability Zone or subnet in a specified list

The following examples specify two launch configurations with different Availability Zones or subnets, but the same instance type and AMI.

Availability Zones

If your account supports EC2-VPC only, the Spot fleet launches the instances in the default subnet of the lowest-priced Availability Zone that you specified. If your account supports EC2-Classic, the Spot fleet launches the instances in the lowest-priced Availability Zone that you specified.

```
{
    "SpotPrice": "0.07",
    "TargetCapacity": 20,
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "SecurityGroups": [
                {
                    "GroupId": "sg-0a42d66a"
                }
            ],
            "InstanceType": "m3.medium",
            "Placement": {
```

```
              "AvailabilityZone": "us-west-2a"
            },
            "IamInstanceProfile": {
               "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"

            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "SecurityGroups": [
                {
                    "GroupId": "sg-0a42d66a"
                }
            ],
            "InstanceType": "m3.medium",
            "Placement": {
              "AvailabilityZone": "us-west-2b"
            },
            "IamInstanceProfile": {
               "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"

            }
        }
    ]
}
```

Subnets

In the launch configurations, you can specify default subnets or nondefault subnets, and the nondefault subnets can be from a default VPC or a nondefault VPC. The Spot service launches the instances in whichever subnet is in the lowest-priced Availability Zone.

Note that you can't specify different subnets from the same Availability Zone in a Spot fleet request.

```
{
    "SpotPrice": "0.07",
    "TargetCapacity": 20,
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "SecurityGroups": [
                {
                    "GroupId": "sg-223b284e"
                }
            ],
            "InstanceType": "m3.medium",
            "SubnetId": "subnet-a61dafcf",
            "IamInstanceProfile": {
               "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"

            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "SecurityGroups": [
                {
```

```
                    "GroupId": "sg-223b284e"
                }
            ],
            "InstanceType": "m3.medium",
            "SubnetId": "subnet-65ea5f08",
            "IamInstanceProfile": {
               "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"

            }
        }
    ]
}
```

Example 3: Launch Spot instances using the lowest-priced instance type in a specified list

The following examples specify two launch configurations with different instance types, but the same AMI and Availability Zone or subnet. The Spot fleet launches the instances using the specified instance type with the lowest price.

Availability Zone

```
{
    "SpotPrice": "2.80",
    "TargetCapacity": 20,
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "SecurityGroups": [
                {
                    "GroupId": "sg-0a42d66a"
                }
            ],
            "InstanceType": "cc2.8xlarge",
            "Placement": {
              "AvailabilityZone": "us-west-2b"
            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "SecurityGroups": [
                {
                    "GroupId": "sg-0a42d66a"
                }
            ],
            "InstanceType": "r3.8xlarge",
            "Placement": {
              "AvailabilityZone": "us-west-2b"
            }
        }
    ]
}
```

Subnet

```
{
    "SpotPrice": "2.80",
    "TargetCapacity": 20,
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "SecurityGroups": [
                {
                    "GroupId": "sg-223b284e"
                }
            ],
            "InstanceType": "cc2.8xlarge",
            "SubnetId": "subnet-482e4972"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "SecurityGroups": [
                {
                    "GroupId": "sg-223b284e"
                }
            ],
            "InstanceType": "r3.8xlarge",
            "SubnetId": "subnet-482e4972"
        }
    ]
}
```

Example 4. Override the Spot price for the request

The ability to specify Spot prices for individual launch specifications provides you with additional control over the bidding process. The following examples override the Spot price for the request (0.070) with individual Spot prices for two of the three launch specifications. Note that the Spot price for the request is used for any launch specification that does not specify an individual Spot price. The Spot fleet launches the instances using the instance type with the lowest price.

Availability Zone

```
{
    "SpotPrice": "1.68",
    "TargetCapacity": 30,
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.2xlarge",
            "Placement": {
              "AvailabilityZone": "us-west-2b"
            },
            "SpotPrice": "0.04"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.4xlarge",
            "Placement": {
              "AvailabilityZone": "us-west-2b"
            },
```

```
            "SpotPrice": "0.06"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.8xlarge",
            "Placement": {
              "AvailabilityZone": "us-west-2b"
            }
        }
    ]
}
```

Subnet

```
{
    "SpotPrice": "1.68",
    "TargetCapacity": 30,
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.2xlarge",
            "SubnetId": "subnet-482e4972",
            "SpotPrice": "0.04"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.4xlarge",
            "SubnetId": "subnet-482e4972",
            "SpotPrice": "0.06"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.8xlarge",
            "SubnetId": "subnet-482e4972"
        }
    ]
}
```

Example 5: Launch a Spot fleet using the diversified allocation strategy

The following example uses the `diversified` allocation strategy. The launch configurations have different instance types but the same AMI and Availability Zone or subnet. The Spot fleet distributes the 30 instances across the 3 launch specifications, such that there are 10 instances of each type. For more information, see Spot Fleet Allocation Strategy (p. 127).

Availability Zone

```
{
    "SpotPrice": "0.70",
    "TargetCapacity": 30,
    "AllocationStrategy": "diversified",
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
```

```
            "InstanceType": "c4.2xlarge",
            "Placement": {
              "AvailabilityZone": "us-west-2b"
            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "m3.2xlarge",
            "Placement": {
              "AvailabilityZone": "us-west-2b"
            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "Placement": {
              "AvailabilityZone": "us-west-2b"
            }
        }
    ]
}
```

Subnet

```
{
    "SpotPrice": "0.70",
    "TargetCapacity": 30,
    "AllocationStrategy": "diversified",
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c4.2xlarge",
            "SubnetId": "subnet-482e4972"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "m3.2xlarge",
            "SubnetId": "subnet-482e4972"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "SubnetId": "subnet-482e4972"
        }
    ]
}
```

Example 6: Launch a Spot fleet using instance weighting

The following examples use instance weighting, which means that the bid price is per unit hour instead of per instance hour. Each launch configuration lists a different instance type and a different weight. The Spot fleet selects the instance type with the lowest price per unit hour. The Spot fleet calculates the number of Spot instances to launch by dividing the target capacity by the instance weight. If the result isn't an integer, the Spot fleet rounds it up to the next integer, so that the size of your fleet is not below its target capacity.

If the `r3.2xlarge` bid is successful, Spot provisions 4 of these instances. (Divide 20 by 6 for a total of 3.33 instances, then round up to 4 instances.)

If the `c3.xlarge` bid is successful, Spot provisions 7 of these instances. (Divide 20 by 3 for a total of 6.66 instances, then round up to 7 instances.)

For more information, see Spot Fleet Instance Weighting (p. 128).

Availability Zone

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
    "LaunchSpecifications": [
      {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "r3.2xlarge",
        "Placement": {
          "AvailabilityZone": "us-west-2b"
        },
        "WeightedCapacity": 6
      },
      {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "c3.xlarge",
        "Placement": {
          "AvailabilityZone": "us-west-2b"
        },
        "WeightedCapacity": 3
      }
    ]
}
```

Subnet

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
    "LaunchSpecifications": [
      {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "r3.2xlarge",
        "SubnetId": "subnet-482e4972",
        "WeightedCapacity": 6
      },
      {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "c3.xlarge",
        "SubnetId": "subnet-482e4972",
        "WeightedCapacity": 3
      }
  ]
}
```

Priority

You can also use instance weighting to give priority to an Availability Zone or subnet. For example, the following launch specifications are nearly identical, except that they specify different subnets and weights. The Spot fleet finds the specification with the highest value for `WeightedCapacity`, and attempts to provision the request in the least expensive Spot pool in that subnet. (Note that the second launch specification does not include a weight, so it defaults to 1.)

```
{
  "SpotPrice": "0.42",
  "TargetCapacity": 40,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
    "LaunchSpecifications": [
      {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "c3.2xlarge",
        "SubnetId": "subnet-482e4972",
        "WeightedCapacity": 2
      },
      {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "c3.2xlarge",
        "SubnetId": "subnet-bb3337d"
      }
    ]
}
```

Spot Bid Status

To help you track your Spot instance requests, plan your use of Spot instances, and bid strategically, Amazon EC2 provides a *bid status*. For example, a bid status can tell you the reason why your Spot request isn't fulfilled yet, or list the constraints that are preventing the fulfillment of your Spot request.

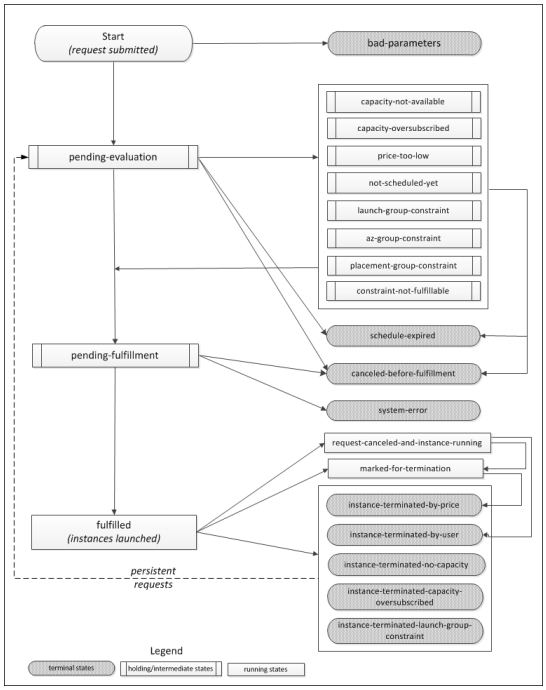
At each step of the process—also called the Spot request *life cycle*, specific events determine successive request states.

Contents

- Life Cycle of a Spot Request (p. 150)
- Getting Bid Status Information (p. 153)
- Spot Bid Status Codes (p. 154)

Life Cycle of a Spot Request

The following diagram shows you the paths that your Spot request can follow throughout its life cycle, from submission to termination. Each step is depicted as a node, and the status code for each node describes the status of the Spot request and Spot instance.



Pending evaluation

As soon as you make a Spot instance request, it goes into the `pending-evaluation` state unless one or more request parameters is not valid (`bad-parameters`).

Status Code	Request State	Instance State
`pending-evaluation`	`open`	n/a
`bad-parameters`	`closed`	n/a

Holding

If one or more request constraints are valid but can't be met yet, or if there is not enough capacity, the request goes into a holding state waiting for the constraints to be met. The request options affect the likelihood of the request being fulfilled. For example, if you specify a bid price below the current Spot price, your request stays in a holding state until the Spot price goes below your bid price. If you specify an Availability Zone group, the request stays in a holding state until the Availability Zone constraint is met.

Status Code	Request State	Instance State
`capacity-not-available`	`open`	n/a
`capacity-oversubscribed`	`open`	n/a
`price-too-low`	`open`	n/a
`not-scheduled-yet`	`open`	n/a
`launch-group-constraint`	`open`	n/a
`az-group-constraint`	`open`	n/a

Status Code	Request State	Instance State
placement-group-constraint	open	n/a
constraint-not-fulfillable	open	n/a

Pending evaluation/fulfillment-terminal

Your Spot instance request can go to a `terminal` state if you create a request that is valid only during a specific time period and this time period expires before your request reaches the pending fulfillment phase, you cancel the request, or a system error occurs.

Status Code	Request State	Instance State
schedule-expired	closed	n/a
canceled-before-fulfillment*	canceled	n/a
bad-parameters	failed	n/a
system-error	closed	n/a

* If you cancel the request.

Pending fulfillment

When the constraints you specified (if any) are met and your bid price is equal to or higher than the current Spot price, your Spot request goes into the `pending-fulfillment` state.

At this point, Amazon EC2 is getting ready to provision the instances that you requested. If the process stops at this point, it is likely to be because it was canceled by the user before a Spot instance was launched, or because an unexpected system error occurred.

Status Code	Request State	Instance State
pending-fulfillment	open	n/a

Fulfilled

When all the specifications for your Spot instances are met, your Spot request is fulfilled. Amazon EC2 launches the Spot instances, which can take a few minutes.

Status Code	Request State	Instance State
fulfilled	active	pending → running

Fulfilled-terminal

Your Spot instances continue to run as long as your bid price is at or above the Spot price, there is spare Spot capacity for your instance type, and you don't terminate the instance. If a change in Spot price or available capacity requires Amazon EC2 to terminate your Spot instances, the Spot request goes into a terminal state. For example, if your bid equals the Spot price but Spot instances are oversubscribed at

that price, the status code is `instance-terminated-capacity-oversubscribed`. A request also goes into the terminal state if you cancel the Spot request or terminate the Spot instances.

Status Code	Request State	Instance State
`request-canceled-and-instance-running`	`canceled`	`running`
`marked-for-termination`	`closed`	`running`
`instance-terminated-by-price`	`closed` (one-time), open (persistent)	`terminated`
`instance-terminated-by-user`	`closed` or `canceled` *	`terminated`
`instance-terminated-no-capacity`	`closed` (one-time), open (persistent)	`terminated`
`instance-terminated-capacity-oversubscribed`	`closed` (one-time), open (persistent)	`terminated`
`instance-terminated-launch-group-constraint`	`closed` (one-time), open (persistent)	`terminated`

* The request state is `closed` if you terminate the instance but do not cancel the bid. The request state is `canceled` if you terminate the instance and cancel the bid. Note that even if you terminate a Spot instance before you cancel its request, there might be a delay before Amazon EC2 detects that your Spot instance was terminated. In this case, the request state can either be `closed` or `canceled`.

Persistent requests

When your Spot instances are terminated (either by you or Amazon EC2), if the Spot request is a persistent request, it returns to the `pending-evaluation` state and then Amazon EC2 can launch a new Spot instance when the constraints are met.

Getting Bid Status Information

You can get bid status information using the AWS Management Console or a command line tool.

To get bid status information using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Spot Requests**, and then select the Spot request.
3. Check the value of **Status** in the **Description** tab.

To get bid status information using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-spot-instance-requests (AWS CLI)
- ec2-describe-spot-instance-requests (Amazon EC2 CLI)
- Get-EC2SpotInstanceRequest (AWS Tools for Windows PowerShell)

Spot Bid Status Codes

Spot bid status information is composed of a bid status code, the update time, and a status message. Together, they help you determine the disposition of your Spot request.

The following list describes the Spot bid status codes:

`az-group-constraint`
: Amazon EC2 cannot launch all the instances you requested in the same Availability Zone.

`bad-parameters`
: One or more parameters for your Spot request are not valid (for example, the AMI you specified does not exist). The bid status message indicates which parameter is not valid.

`canceled-before-fulfillment`
: The user canceled the Spot request before it was fulfilled.

`capacity-not-available`
: There is not enough capacity available for the instances that you requested.

`capacity-oversubscribed`
: The number of Spot requests with bid prices equal to or higher than your bid price exceeds the available capacity in this Spot pool.

`constraint-not-fulfillable`
: The Spot request can't be fulfilled because one or more constraints are not valid (for example, the Availability Zone does not exist). The bid status message indicates which constraint is not valid.

`fulfilled`
: The Spot request is `active`, and Amazon EC2 is launching your Spot instances.

`instance-terminated-by-price`
: The Spot price rose above your bid price. If your request is a persistent bid, the process restarts, so your bid is pending evaluation.

`instance-terminated-by-user` or `spot-instance-terminated-by-user`
: You terminated a Spot instance that had been fulfilled, so the bid state is `closed` (unless it's a persistent bid) and the instance state is `terminated`.

`instance-terminated-capacity-oversubscribed`
: Your instance is terminated because the number of Spot requests with bid prices equal to or higher than your bid price exceeded the available capacity in this Spot pool. (Note that the Spot price might not have changed.) The Spot service randomly selects instances to be terminated.

`instance-terminated-launch-group-constraint`
: One or more of the instances in your launch group was terminated, so the launch group constraint is no longer fulfilled.

`instance-terminated-no-capacity`
: There is no longer enough Spot capacity available for the instance.

`launch-group-constraint`
: Amazon EC2 cannot launch all the instances that you requested at the same time. All instances in a launch group are started and terminated together.

`marked-for-termination`
: The Spot instance is marked for termination.

`not-scheduled-yet`
: The Spot request will not be evaluated until the scheduled date.

`pending-evaluation`
: After you make a Spot instance request, it goes into the `pending-evaluation` state while the system evaluates the parameters of your request.

`pending-fulfillment`
: Amazon EC2 is trying to provision your Spot instances.

`placement-group-constraint`
: The Spot request can't be fulfilled yet because a Spot instance can't be added to the placement group at this time.

`price-too-low`
: The bid request can't be fulfilled yet because the bid price is below the Spot price. In this case, no instance is launched and your bid remains `open`.

`request-canceled-and-instance-running`
: You canceled the Spot request while the Spot instances are still running. The request is `canceled`, but the instances remain `running`.

`schedule-expired`
: The Spot request expired because it was not fulfilled before the specified date.

`system-error`
: There was an unexpected system error. If this is a recurring issue, please contact customer support for assistance.

Spot Instance Interruptions

Demand for Spot instances can vary significantly from moment to moment, and the availability of Spot instances can also vary significantly depending on how many unused EC2 instances are available. In addition, no matter how high you bid, it is still possible that your Spot instance will be interrupted. Therefore, you must ensure that your application is prepared for a Spot instance interruption. We strongly recommend that you do not use Spot instances for applications that can't be interrupted.

The following are the possible reasons that Amazon EC2 will terminate your Spot instances:

- Price—The Spot price is greater than your bid price.
- Capacity—If there are not enough unused EC2 instances to meet the demand for Spot instances, Amazon EC2 terminates Spot instances, starting with those instances with the lowest bid prices. If there are several Spot instances with the same bid price, the order in which the instances are terminated is determined at random.
- Constraints—If your request includes a constraint such as a launch group or an Availability Zone group, these Spot instances are terminated as a group when the constraint can no longer be met.

Preparing for Interruptions

Here are some best practices to follow when you use Spot instances:

- Choose a reasonable bid price. Your bid price should be high enough to make it likely that your request will be fulfilled, but not higher than you are willing to pay. This is important because if the supply is low for an extended period of time, the Spot price can remain high during that period because it is based on the highest bid prices. We strongly recommend against bidding above the price for On-Demand instances.
- Ensure that your instance is ready to go as soon as the request is fulfilled by using an Amazon Machine Image (AMI) that contains the required software configuration. You can also use user data to run commands at start-up.
- Store important data regularly in a place that won't be affected when the Spot instance terminates. For example, you can use Amazon S3, Amazon EBS, or DynamoDB.
- Divide the work into small tasks (using a Grid, Hadoop, or queue-based architecture) or use checkpoints so that you can save your work frequently.
- Use Spot instance termination notices to monitor the status of your Spot instances.
- Test your application to ensure that it handles an unexpected instance termination gracefully. You can do so by running the application using an On-Demand instance and then terminating the On-Demand instance yourself.

Spot Instance Termination Notices

The best way to protect against Spot instance interruption is to architect your application to be fault tolerant. In addition, you can take advantage of *Spot instance termination notices*, which provide a two-minute warning before Amazon EC2 must terminate your Spot instance.

This warning is made available to the applications on your Spot instance using an item in the instance metadata. For example, you can check for this warning in the instance metadata periodically (we recommend every 5 seconds) using the following query:

```
C:\> invoke-restmethod -uri http://169.254.169.254/latest/meta-data/spot/termination-time
```

For information about other ways to retrieve instance metadata, see Retrieving Instance Metadata (p. 161).

If your Spot instance is marked for termination by Amazon EC2, the `termination-time` item is present and it specifies the approximate time in UTC when the instance will receive the shutdown signal. For example:

```
2015-01-05T18:02:00Z
```

If Amazon EC2 is not preparing to terminate the instance, or if you terminated the Spot instance yourself, the `termination-time` item is either not present (so you receive an HTTP 404 error) or contains a value that is not a time value.

Note that while we make every effort to provide this warning the moment that your Spot instance is marked for termination by Amazon EC2, it is possible that your Spot instance will be terminated before Amazon EC2 can make the warning available. Therefore, you must ensure that your application is prepared to handle an unexpected Spot instance interruption even if you are checking for Spot instance termination notices.

Spot Instance Data Feed

To help you understand the charges for your Spot instances, Amazon EC2 provides a data feed that describes your Spot instance usage and pricing. This data feed is sent to an Amazon S3 bucket that you specify when you subscribe to the data feed.

Data feed files arrive in your bucket typically once an hour, and each hour of usage is typically covered in a single data file. These files are compressed (gzip) before they are delivered to your bucket. Amazon EC2 can write multiple files for a given hour of usage where files are very large (for example, when file contents for the hour exceed 50 MB before compression).

> **Note**
> If you don't have a Spot instance running during a certain hour, you won't receive a data feed file for that hour.

Contents

- Data Feed File Name and Format (p. 157)
- Amazon S3 Bucket Permissions (p. 157)
- Subscribing to Your Spot instance Data Feed (p. 158)
- Deleting Your Spot Instance Data Feed (p. 158)

Data Feed File Name and Format

The Spot instance data feed file name uses the following format (with the date and hour in UTC):

```
bucket-name.s3.amazonaws.com/{optional prefix}/aws-account-id.YYYY-MM-DD-HH.n.unique-id.gz
```

For example, if your bucket name is `myawsbucket` and your prefix is `myprefix`, your file names are similar to the following:

```
myawsbucket.s3.amazonaws.com/myprefix/111122223333.2014-03-17-20.001.pwBdGTJG.gz
```

The Spot instance data feed files are tab-delimited. Each line in the data file corresponds to one instance hour and contains the fields listed in the following table.

Field	Description
`Timestamp`	The timestamp used to determine the price charged for this instance hour.
`UsageType`	The type of usage and instance type being charged for. For `m1.small` Spot instances, this field is set to `SpotUsage`. For all other instance types, this field is set to `SpotUsage:`{*instance-type*}. For example, `SpotUsage:c1.medium`.
`Operation`	The product being charged for. For Linux Spot instances, this field is set to `RunInstances`. For Microsoft Windows Spot instances, this field is set to `RunInstances:0002`. Spot usage is grouped according to Availability Zone.
`InstanceID`	The ID of the Spot instance that generated this instance hour.
`MyBidID`	The ID for the Spot instance request that generated this instance hour.
`MyMaxPrice`	The maximum price specified for this Spot instance request.
`MarketPrice`	The Spot price at the time specified in the `Timestamp` field.
`Charge`	The price charged for this instance hour.
`Version`	The version included in the data feed file name for this record.

Amazon S3 Bucket Permissions

When you subscribe to the data feed, you must specify an Amazon S3 bucket to store the data feed files. Before you choose an Amazon S3 bucket for the data feed, consider the following:

- You must have `FULL_CONTROL` permission to the bucket.

  If you're the bucket owner, you have this permission by default. Otherwise, the bucket owner must grant your AWS account this permission.
- When you create your data feed subscription, Amazon S3 updates the ACL of the specified bucket to allow the AWS data feed account read and write permissions.
- Removing the permissions for the data feed account does not disable the data feed. If you remove those permissions but don't disable the data feed, we restore those permissions the next time that the data feed account needs to write to the bucket.
- Each data feed file has its own ACL (separate from the ACL for the bucket). The bucket owner has `FULL_CONTROL` permission to the data files. The data feed account has read and write permissions.

- If you delete your data feed subscription, Amazon EC2 doesn't remove the read and write permissions for the data feed account on either the bucket or the data files. You must remove these permissions yourself.

Subscribing to Your Spot instance Data Feed

You can subscribe to your Spot instance data feed using the command line or API.

Subscribe Using the AWS CLI

To subscribe to your data feed, use the following create-spot-datafeed-subscription command:

```
C:\> aws ec2 create-spot-datafeed-subscription --bucket myawsbucket [--prefix
myprefix]
```

The following is example output:

```
{
    "SpotDatafeedSubscription": {
        "OwnerId": "111122223333",
        "Prefix": "myprefix",
        "Bucket": "myawsbucket",
        "State": "Active"
    }
}
```

Subscribe Using the Amazon EC2 CLI

To subscribe to your data feed, use the following ec2-create-spot-datafeed-subscription command:

```
C:\> ec2-create-spot-datafeed-subscription --bucket myawsbucket [--prefix
myprefix]
```

The following is example output:

```
SPOTDATAFEEDSUBSCRIPTION        111122223333    myawsbucket        myprefix
Active
```

Deleting Your Spot Instance Data Feed

You can delete your Spot instance data feed using the command line or API .

Delete Using the AWS CLI

To delete your data feed, use the following delete-spot-datafeed-subscription command:

```
C:\> aws ec2 delete-spot-datafeed-subscription
```

Delete Using the Amazon EC2 CLI

To delete your data feed, use the following ec2-delete-spot-datafeed-subscription command:

```
C:\> ec2-delete-spot-datafeed-subscription
```

Spot Instance Limits

Spot instance requests are subject to the following limits:

Limits

- Overall Spot Request Limit (p. 159)
- Unsupported Instance Types (p. 159)
- Spot Bid Price Limit (p. 159)
- Spot Fleet Limits (p. 160)

MaxSpotInstanceCountExceeded Error

If you submit a Spot instance request and you receive the error `Max spot instance count exceeded`, your account has exceeded either its overall limit for the region, or the limit for the specific instance type for the region. To submit a limit increase request, go to AWS Support Center and complete the request form. In the **Use Case Description** field, indicate that you are requesting an increase to the request limit for Spot instances.

Overall Spot Request Limit

The overall limit applies to active or open Spot instance requests. If you terminate your Spot instance but do not cancel the request, your overall limit can include the request until Amazon EC2 detects the termination and closes your request.

The following table lists the overall request limit for Spot instances. Note that new AWS accounts might have lower limits.

Resource	Limit
The total number of Spot instance requests	20 per region

Unsupported Instance Types

The following instance types are not supported for Spot:

- T2
- I2
- HS1

Some Spot instance types aren't available in every region. To view the supported instance types for a region, go to Spot Instance Pricing and select the region from the **Region** list.

Spot Bid Price Limit

The bid price limit is designed to protect you from incurring unexpected charges.

The following table lists the bid price limit for Spot instances.

Resource	Limit
Bid price	Ten times the On-Demand price

Spot Fleet Limits

The usual Amazon EC2 limits apply to instances launched by a Spot fleet, such as Spot bid price limits, instance limits, and volume limits. In addition, the following limits apply:

- The number of active Spot fleets per region: 1,000
- The number of launch specifications per fleet: 20
- The size of the user data in a launch specification: 16 KB
- The number of instances per Spot fleet: 3,000
- The number of instances in all Spot fleets in a region: 3,000
- A Spot fleet request can't span regions.
- A Spot fleet request can't span different subnets from the same Availability Zone.

Instance Metadata and User Data

Instance metadata is data about your instance that you can use to configure or manage the running instance. Instance metadata is divided into categories. For more information, see Instance Metadata Categories (p. 164).

EC2 instances can also include *dynamic data*, such as an instance identity document that is generated when the instance is launched. For more information, see Dynamic Data Categories (p. 168).

You can also access the *user data* that you supplied when launching your instance. For example, you can specify parameters for configuring your instance, or attach a simple script. You can also use this data to build more generic AMIs that can be modified by configuration files supplied at launch time. For example, if you run web servers for various small businesses, they can all use the same AMI and retrieve their content from the Amazon S3 bucket you specify in the user data at launch. To add a new customer at any time, simply create a bucket for the customer, add their content, and launch your AMI. If you launch more than one instance at the same time, the user data is available to all instances in that reservation.

Important
Although you can only access instance metadata and user data from within the instance itself, the data is not protected by cryptographic methods. Anyone who can access the instance can view its metadata. Therefore, you should take suitable precautions to protect sensitive data (such as long-lived encryption keys). You should not store sensitive data, such as passwords, as user data.

Contents

- Retrieving Instance Metadata (p. 161)
- Adding User Data (p. 163)
- Retrieving User Data (p. 163)
- Retrieving Dynamic Data (p. 164)
- Instance Metadata Categories (p. 164)

Retrieving Instance Metadata

Because your instance metadata is available from your running instance, you do not need to use the Amazon EC2 console or the AWS CLI. This can be helpful when you're writing scripts to run from your instance. For example, you can access the local IP address of your instance from instance metadata to manage a connection to an external application.

To view all categories of instance metadata from within a running instance, use the following URI:

```
http://169.254.169.254/latest/meta-data/
```

Note that you are not billed for HTTP requests used to retrieve instance metadata and user data.

You can install a tool such as GNU Wget or cURL to retrieve instance metadata at the command line, or you can copy and paste the URI into a browser. If you do not want to install any third-party tools, you can use PowerShell cmdlets to retrieve the URI. For example, if you are running version 3.0 or later of PowerShell, use the following cmdlet:

```
PS C:\> invoke-restmethod -uri http://169.254.169.254/latest/meta-data/
```

Important
If you do install a third-party tool on a Windows instance, ensure that you read the accompanying documentation carefully, as the method of calling the HTTP and the output format might be different from what is documented here.

All metadata is returned as text (content type text/plain). A request for a specific metadata resource returns the appropriate value, or a `404 - Not Found` HTTP error code if the resource is not available.

A request for a general metadata resource (the URI ends with a /) returns a list of available resources, or a `404 - Not Found` HTTP error code if there is no such resource. The list items are on separate lines, terminated by line feeds (ASCII 10).

Examples of Retrieving Instance Metadata

This example gets the available versions of the instance metadata. These versions do not necessarily correlate with an Amazon EC2 API version. The earlier versions are available to you in case you have scripts that rely on the structure and information present in a previous version.

```
C:\> curl http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
latest
```

This example gets the top-level metadata items. Some items are only available for instances in a VPC. For more information about each of these items, see Instance Metadata Categories (p. 164).

```
C:\> curl http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
instance-action
instance-id
instance-type
kernel-id
local-hostname
local-ipv4
mac
network/
placement/
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
```

These examples get the value of some of the metadata items from the preceding example.

```
C:\> curl http://169.254.169.254/latest/meta-data/ami-id
ami-2bb65342
```

```
C:\> curl http://169.254.169.254/latest/meta-data/reservation-id
r-fea54097
```

```
C:\> curl http://169.254.169.254/latest/meta-data/hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

This example shows the information available for a specific network interface (indicated by the MAC address) on an NAT instance in the EC2-Classic platform.

```
C:\> curl http://169.254.169.254/latest/meta-data/network/inter
faces/macs/02:29:96:8f:6a:2d/
device-number
local-hostname
local-ipv4s
mac
owner-id
public-hostname
public-ipv4s
```

This example gets the subnet ID for an instance launched into a VPC.

```
C:\> curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

Adding User Data

When you specify user data, note the following:

- User data is treated as opaque data: what you give is what you get back. It is up to the instance to be able to interpret it.
- User data is limited to 16 KB. This limit applies to the data in raw form, not base64-encoded form.
- User data must be base64-encoded before being submitted to the API. The EC2 command line tools perform the base64 encoding for you. The data is decoded before being presented to the instance. For more information about base64 encoding, see http://tools.ietf.org/html/rfc4648.
- User data is executed only at launch. If you stop an instance, modify the user data, and start the instance, the new user data is not executed automatically.

To specify user data when you launch an instance

You can specify user data when you launch an instance. For more information, see Launching an Instance (p. 207).

To modify the user data for an Amazon EBS-backed instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Instances**, and select the instance.
3. Click **Actions**, select **Instance State**, and then click **Stop**.

   Warning
   When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.
4. In the confirmation dialog box, click **Yes, Stop**. It can take a few minutes for the instance to stop.
5. With the instance still selected, click **Actions**, select **Instance Settings**, and then click **View/Change User Data**. Note that you can't change the user data if the instance is running, but you can view it.
6. In the **View/Change User Data** dialog box, update the user data, and then click **Save**.

After you modify the user data for your instance, you can execute it. For more information, see Executing User Data (p. 243).

Retrieving User Data

To retrieve user data, use the following URI:

```
http://169.254.169.254/latest/user-data
```

Requests for user data returns the data as it is (content type application/x-octetstream).

This shows an example of returning comma-separated user data.

```
C:\> curl http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

This shows an example of returning line-separated user data.

```
C:\> curl http://169.254.169.254/latest/user-data
[general]
instances: 4

[instance-0]
s3-bucket: <user_name>

[instance-1]
reboot-on-error: yes
```

Retrieving Dynamic Data

To retrieve dynamic data from within a running instance, use the following URI:

```
http://169.254.169.254/latest/dynamic/
```

This example shows how to retrieve the high-level instance identity categories:

```
C:\> curl http://169.254.169.254//latest/dynamic/instance-identity/
pkcs7
signature
document
```

Instance Metadata Categories

The following table lists the categories of instance metadata.

Data	Description	Version Introduced
`ami-id`	The AMI ID used to launch the instance.	1.0
`ami-launch-index`	If you started more than one instance at the same time, this value indicates the order in which the instance was launched. The value of the first instance launched is 0.	1.0
`ami-manifest-path`	The path to the AMI's manifest file in Amazon S3. If you used an Amazon EBS-backed AMI to launch the instance, the returned result is `unknown`.	1.0

Data	Description	Version Introduced
`ancestor-ami-ids`	The AMI IDs of any instances that were rebundled to create this AMI. This value will only exist if the AMI manifest file contained an `ancestor-amis` key.	2007-10-10
`block-device-mapping/ami`	The virtual device that contains the root/boot file system.	2007-12-15
`block-device-mapping/ebs`*N*	The virtual devices associated with Amazon EBS volumes, if any are present. Amazon EBS volumes are only available in metadata if they were present at launch time or when the instance was last started. The *N* indicates the index of the Amazon EBS volume (such as `ebs1` or `ebs2`).	2007-12-15
`block-device-mapping/ephemeral`*N*	The virtual devices associated with ephemeral devices, if any are present. The *N* indicates the index of the ephemeral volume.	2007-12-15
`block-device-mapping/root`	The virtual devices or partitions associated with the root devices, or partitions on the virtual device, where the root (/ or C:) file system is associated with the given instance.	2007-12-15
`block-device-mapping/swap`	The virtual devices associated with `swap`. Not always present.	2007-12-15
`hostname`	The private hostname of the instance. In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0).	1.0
`iam/info`	If there is an IAM role associated with the instance at launch, contains information about the last time the instance profile was updated, including the instance's LastUpdated date, InstanceProfileArn, and InstanceProfileId. Otherwise, not present.	2012-01-12
`iam/security-credentials/role-name`	If there is an IAM role associated with the instance at launch, *`role-name`* is the name of the role, and *`role-name`* contains the temporary security credentials associated with the role (for more information, see Retrieving Security Credentials from Instance Metadata (p. 443)). Otherwise, not present.	2012-01-12

Data	Description	Version Introduced
`instance-action`	Notifies the instance that it should reboot in preparation for bundling. Valid values: `none` | `shutdown` | `bundle-pending`.	2008-09-01
`instance-id`	The ID of this instance.	1.0
`instance-type`	The type of instance. For more information, see Instance Types (p. 96).	2007-08-29
`kernel-id`	The ID of the kernel launched with this instance, if applicable.	2008-02-01
`local-hostname`	The private DNS hostname of the instance. In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0).	2007-01-19
`local-ipv4`	The private IP address of the instance. In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0).	1.0
`mac`	The instance's media access control (MAC) address. In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0).	2011-01-01
`network/interfaces/macs/mac/device-number`	The unique device number associated with that interface. The device number corresponds to the device name; for example, a `device-number` of 2 is for the eth2 device. This category corresponds to the `DeviceIndex` and `device-index` fields that are used by the Amazon EC2 API, the Amazon EC2 CLI, and the EC2 commands for the AWS CLI.	2011-01-01
`network/interfaces/macs/mac/ipv4-associations/public-ip`	The private IPv4 addresses that are associated with each `public-ip` address and assigned to that interface.	2011-01-01
`network/interfaces/macs/mac/local-hostname`	The interface's local hostname.	2011-01-01
`network/interfaces/macs/mac/local-ipv4s`	The private IP addresses associated with the interface.	2011-01-01
`network/interfaces/macs/mac/mac`	The instance's MAC address.	2011-01-01

Data	Description	Version Introduced
`network/interfaces/macs/mac/owner-id`	The ID of the owner of the network interface. In multiple-interface environments, an interface can be attached by a third party, such as Elastic Load Balancing. Traffic on an interface is always billed to the interface owner.	2011-01-01
`network/interfaces/macs/mac/public-hostname`	The interface's public DNS. If the instance is in a VPC, this category is only returned if the `enableDnsHostnames` attribute is set to `true`. For more information, see Using DNS with Your VPC.	2011-01-01
`network/interfaces/macs/mac/public-ipv4s`	The Elastic IP addresses associated with the interface. There may be multiple IP addresses on an instance.	2011-01-01
`network/interfaces/macs/mac/security-groups`	Security groups to which the network interface belongs. Returned only for instances launched into a VPC.	2011-01-01
`network/interfaces/macs/mac/security-group-ids`	IDs of the security groups to which the network interface belongs. Returned only for instances launched into a VPC. For more information on security groups in the EC2-VPC platform, see Security Groups for Your VPC.	2011-01-01
`network/interfaces/macs/mac/subnet-id`	The ID of the subnet in which the interface resides. Returned only for instances launched into a VPC.	2011-01-01
`network/interfaces/macs/mac/subnet-ipv4-cidr-block`	The CIDR block of the subnet in which the interface resides. Returned only for instances launched into a VPC.	2011-01-01
`network/interfaces/macs/mac/vpc-id`	The ID of the VPC in which the interface resides. Returned only for instances launched into a VPC.	2011-01-01
`network/interfaces/macs/mac/vpc-ipv4-cidr-block`	The CIDR block of the VPC in which the interface resides. Returned only for instances launched into a VPC.	2011-01-01
`placement/availability-zone`	The Availability Zone in which the instance launched.	2008-02-01
`product-codes`	Product codes associated with the instance, if any.	2007-03-01
`public-hostname`	The instance's public DNS. If the instance is in a VPC, this category is only returned if the `enableDnsHostnames` attribute is set to `true`. For more information, see Using DNS with Your VPC.	2007-01-19

Data	Description	Version Introduced
`public-ipv4`	The public IP address. If an Elastic IP address is associated with the instance, the value returned is the Elastic IP address.	2007-01-19
`public-keys/0/openssh-key`	Public key. Only available if supplied at instance launch time.	1.0
`ramdisk-id`	The ID of the RAM disk specified at launch time, if applicable.	2007-10-10
`reservation-id`	The ID of the reservation.	1.0
`security-groups`	The names of the security groups applied to the instance. After launch, you can only changes the security groups of instances running in a VPC. Such changes are reflected here and in network/interfaces/macs/*mac*/security-groups.	1.0
`services/domain`	The domain for AWS resources for the region; for example, `amazonaws.com` for us-east-1.	2014-02-25
`spot/termination-time`	The approximate time, in UTC, that the operating system for your Spot instance will receive the shutdown signal. This item is present and contains a time value (for example, 2015-01-05T18:02:00Z) only if the Spot instance has been marked for termination by Amazon EC2. The termination-time item is not set to a time if you terminated the Spot instance yourself.	2015-01-05

Dynamic Data Categories

The following table lists the categories of dynamic data.

Data	Description	Version introduced
`fws/instance-monitoring`	Value showing whether the customer has enabled detailed one-minute monitoring in CloudWatch. Valid values: `enabled | disabled`	2009-04-04
`instance-identity/document`	JSON containing instance attributes, such as instance-id, private IP address, etc.	2009-04-04
`instance-identity/pkcs7`	Used to verify the document's authenticity and content against the signature.	2009-04-04
`instance-identity/signature`	Data that can be used by other parties to verify its origin and authenticity.	2009-04-04

Importing and Exporting Instances

You can use the Amazon Web Services (AWS) VM Import/Export tools to import virtual machine (VM) images from your local environment into AWS and convert them into ready-to-use Amazon EC2 Amazon machine images (AMIs) or instances. Later, you can export the VM images back to your local environment. VM Import/Export allows you to leverage your existing investments in the VMs that you have built to meet your IT security, configuration management, and compliance requirements by bringing those VMs into Amazon Elastic Compute Cloud (Amazon EC2) as ready-to-use AMIs or instances. VM Import/Export is compatible with Citrix Xen, Microsoft Hyper-V, or VMware vSphere virtualization environments. If you're using VMware vSphere, you can also use the AWS Connector for vCenter to export a VM from VMware and import it into Amazon EC2. For more information, see Migrating Your Virtual Machine to Amazon EC2 Using AWS Connector for vCenter in the *AWS Management Portal for vCenter User Guide*. If you use Microsoft Systems Center, you can also use AWS Systems Manager for Microsoft SCVMM to import Windows VMs from SCVMM to Amazon EC2. For more information, see Importing Your Virtual Machine Using AWS Systems Manager for Microsoft SCVMM in the *Amazon EC2 User Guide for Microsoft Windows Instances*.

VM Import/Export can be used to migrate applications and workloads, copy your VM image catalog, or create a disaster recovery repository for VM images.

- **Migrate existing applications and workloads to Amazon EC2**—You can migrate your VM-based applications and workloads to Amazon EC2 and preserve their software and configuration settings. When you import a VM using VM Import, you can convert an existing VM into an Amazon EC2 instance or an Amazon Machine Image (AMI) that you can run on Amazon EC2. When you create an AMI from your VM, you can run multiple instances based on the same imported VM. You can also use the AMI to replicate your applications and workloads around the world using AMI Copy. For more information, see Copying an AMI (p. 74).
- **Import your VM image catalog to Amazon EC2**—You can import your existing VM image catalog into Amazon EC2. If you maintain a catalog of approved VM images, you can copy your image catalog to Amazon EC2 and create AMIs from the imported VM images. Your existing software, including products that you have installed such as anti-virus software, intrusion detection systems, and so on, can be imported along with your VM images. You can use the AMIs you have created as your Amazon EC2 image catalog.
- **Create a disaster recovery repository for VM images**—You can import your local VM images into Amazon EC2 for backup and disaster recovery purposes. You can import your VMs and store them as AMIs. The AMIs you create will be ready to launch in Amazon EC2 when you need them. If your local environment suffers an event, you can quickly launch your instances to preserve business continuity while simultaneously exporting them to rebuild your local infrastructure.

Contents

- VM Import/Export Prerequisites (p. 169)
- Importing a VM into Amazon EC2 Using ImportImage (p. 177)
- Importing a VM into Amazon EC2 Using ImportInstance (p. 185)
- Exporting Amazon EC2 Instances (p. 195)
- Troubleshooting VM Import/Export (p. 197)

VM Import/Export Prerequisites

Before you begin the process of exporting a VM from your virtualization environment or importing and exporting a VM from Amazon EC2, you must be aware of the operating systems and image formats that AWS supports, and understand the limitations on exporting instances and volumes.

To import or export a VM from Amazon EC2, you must also install the CLI tools:

- For more information about installing the Amazon EC2 CLI, see the *Amazon EC2 Command Line Reference*.
- For more information about installing the AWS CLI, see the *AWS Command Line Interface User Guide*. For more information about the Amazon EC2 commands in the AWS CLI, see ec2 in the *AWS Command Line Interface Reference*.

Contents

- Operating Systems (p. 170)
- Image Formats (p. 171)
- Instance Types (p. 171)
- Volume Types and Filesystems (p. 172)
- VM Import Service Role (p. 172)
- IAM Permissions (p. 173)
- Requirements and Limitations (p. 174)

Operating Systems

The following operating systems can be imported into and exported from Amazon EC2.

Windows (32- and 64-bit)

- Microsoft Windows Server 2003 (Standard, Datacenter, Enterprise) with Service Pack 1 (SP1) or later
- Microsoft Windows Server 2003 R2 (Standard, Datacenter, Enterprise)
- Microsoft Windows Server 2008 (Standard, Datacenter, Enterprise)
- Microsoft Windows Server 2008 R2 (Standard, Datacenter, Enterprise)
- Microsoft Windows Server 2012 (Standard, Datacenter)
- Microsoft Windows Server 2012 R2 (Standard, Datacenter)
- Microsoft Windows 7 (Professional, Enterprise, Ultimate)

  Note
  VM Import currently supports importing VMs running US English versions of Microsoft Windows 7 (Professional, Enterprise, Ultimate). When importing these operating systems, you must comply with the Requirements and Limitations (p. 174).

- Microsoft Windows 8 (Professional, Enterprise)

  Note
  VM Import currently supports importing VMs running US English versions of Microsoft Windows 8 (Professional, Enterprise). When importing these operating systems, you must comply with the Requirements and Limitations (p. 174).

- Microsoft Windows 8.1 (Professional, Enterprise)

  Note
  VM Import currently supports importing VMs running US English versions of Microsoft Windows 8.1 (Professional, Enterprise). When importing these operating systems, you must comply with the Requirements and Limitations (p. 174).

Linux/Unix (64-bit)

- Red Hat Enterprise Linux (RHEL) 5.1-5.11, 6.1-6.6, 7.0-7.1

  Note
  RHEL 6.0 is unsupported because it lacks the drivers required to run on Amazon EC2.

VM Import supports license portability for RHEL instances. Your existing RHEL licenses are imported along with their associated RHEL instance. For more information about eligibility for Red Hat Cloud Access, see Eligibility at the Red Hat website.

- CentOS 5.1-5.11, 6.1-6.6, 7.0-7.1

  Note
  CentOS 6.0 is unsupported because it lacks the drivers required to run on Amazon EC2.

- Ubuntu 12.04, 12.10, 13.04, 13.10, 14.04, 14.10
- Debian 6.0.0-6.0.8, 7.0.0-7.2.0

Image Formats

The following formats can be imported into and exported from Amazon EC2.

Importing Image Formats into Amazon EC2

AWS supports the following image formats for importing both disks and VMs into Amazon EC2:

- RAW format for importing disks and VMs.
- Dynamic Virtual Hard Disk (VHD) image formats, which are compatible with Microsoft Hyper-V and Citrix Xen virtualization products. VHDX images are not currently supported.
- Stream-optimized ESX Virtual Machine Disk (VMDK) image format, which is compatible with VMware ESX and VMware vSphere virtualization products.

  Note
  You can only import VMDK files into Amazon EC2 that were created through the OVF export process in VMware.

- Open Virtual Appliance (OVA) image format, which supports importing images with multiple hard disks.

Exporting Image Formats from Amazon EC2

AWS supports the following image formats for exporting both volumes and instances from Amazon EC2. Make sure that you convert your output file to the format that your VM environment supports:

- Open Virtual Appliance (OVA) image format, which is compatible with VMware vSphere versions 4 and 5.
- Virtual Hard Disk (VHD) image format, which is compatible with Citrix Xen and Microsoft Hyper-V virtualization products.
- Stream-optimized ESX Virtual Machine Disk (VMDK) image format, which is compatible with VMware ESX and VMware vSphere versions 4 and 5 virtualization products.

Instance Types

AWS supports importing Windows instances into any instance type. Linux instances can be imported into the following instance types:

- General purpose: `t2.micro` | `t2.small` | `t2.medium` | `m3.medium` | `m3.large` | `m3.xlarge` | `m3.2xlarge`
- Compute optimized: `c3.large` | `c3.xlarge` | `c3.2xlarge` | `c3.4xlarge` | `cc2.8xlarge`
- Memory optimized: `cr1.8xlarge`
- Storage optimized: `hi1.4xlarge` | `hs1.8xlarge` | `i2.xlarge` | `i2.2xlarge` | `i2.4xlarge`
- GPU: `cg1.4xlarge`

Volume Types and Filesystems

AWS supports importing Windows and Linux instances with the following filesystems:

Windows (32- and 64-bit)

VM Import/Export supports MBR-partitioned volumes that are formatted using the NTFS filesystem. GUID Partition Table (GPT) partitioned volumes are not supported.

Linux/Unix (64-bit)

VM Import/Export supports MBR-partitioned volumes that are formatted using ext2, ext3, ext4, Btrfs, JFS, or XFS filesystem. GUID Partition Table (GPT) partitioned volumes are not supported.

VM Import Service Role

VM Import uses a role in your AWS account to perform certain operations (e.g: downloading disk images from an Amazon S3 bucket). You must create a role with the name **vmimport** with the following policy and trusted entities. Create a file named **trust-policy.json** with the following policy:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"",
         "Effect":"Allow",
         "Principal":{
            "Service":"vmie.amazonaws.com"
         },
         "Action":"sts:AssumeRole",
         "Condition":{
            "StringEquals":{
               "sts:ExternalId":"vmimport"
            }
         }
      }
   ]
}
```

Use the `aws iam create-role` command to create a role named **vmimport** and give VM Import/Export access to it.

> **Note**
> The external id must be named **vmimport**.

```
aws iam create-role --role-name vmimport --assume-role-policy-document
file://trust-policy.json
```

> **Note**
> You must include **file://** before the policy document name (e.g., file://trust-policy.json), or the command will return the error "A client error (MalformedPolicyDocument) occurred when calling the CreateRole operation: Syntax errors in policy."

Creating a policy for the service role

Create a file named role-policy.json with the following policy:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:ListBucket",
            "s3:GetBucketLocation"
         ],
         "Resource":[
            "arn:aws:s3:::<disk-image-file-bucket>"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject"
         ],
         "Resource":[
            "arn:aws:s3:::<disk-image-file-bucket>/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "ec2:ModifySnapshotAttribute",
            "ec2:CopySnapshot",
            "ec2:RegisterImage",
            "ec2:Describe*"
         ],
         "Resource":"*"
      }
   ]
}
```

Replace *<disk-image-file-bucket>* with the appropriate Amazon S3 bucket where the disk files are stored. Run the following command to attach the policy to the role created above:

```
aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-
document file://role-policy.json
```

For more information about IAM roles, see IAM Roles (Delegation and Federation) in the *IAM User Guide*.

IAM Permissions

If you're logged on as an AWS Identity and Access Management (IAM) user, you'll need the following permissions in your IAM policy to import or export a VM:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets"
```

```
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:DeleteBucket",
        "s3:DeleteObject",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": ["arn:aws:s3:::mys3bucket","arn:aws:s3:::mys3bucket/*"]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CancelConversionTask",
        "ec2:CancelExportTask",
        "ec2:CreateImage",
        "ec2:CreateInstanceExportTask",
        "ec2:CreateTags",
        "ec2:DeleteTags",
        "ec2:DescribeConversionTasks",
        "ec2:DescribeExportTasks",
        "ec2:DescribeInstanceAttribute",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstances",
        "ec2:DescribeTags",
        "ec2:ImportInstance",
        "ec2:ImportVolume",
        "ec2:StartInstances",
        "ec2:StopInstances",
        "ec2:TerminateInstances",
        "ec2:ImportImage",
        "ec2:ImportSnapshot",
        "ec2:DescribeImportImageTasks",
        "ec2:DescribeImportSnapshotTasks",
        "ec2:CancelImportTask"
      ],
      "Resource": "*"
    }
  ]
}
```

For more information about IAM users and policies, see IAM Users and Groups and Managing IAM Policies in the *IAM User Guide*.

Requirements and Limitations

Known Limitations for Importing a VM into Amazon EC2 Using ImportImage

Importing AMIs and snapshots is subject to the following limitations:

- You can have up to twenty import image or snapshots tasks per region in progress at the same time. To request an increase to this limit, contact AWS Support. Tasks must complete within 7 days of the start date.
- Imported VMs create Amazon EC2 AMIs that use Hardware Virtual Machine (HVM) virtualization. Creating AMIs that use Paravirtual (PV) virtualization using VM Import is not supported. Linux PVHVM drivers are supported within imported instances.
- Imported Red Hat Enterprise Linux (RHEL) instances must use Cloud Access (BYOL) licenses.
- Imported Linux instances must use 64-bit images. Importing 32-bit Linux images is not supported.
- Imported Linux instances should use default kernels for best results. VMs that use custom Linux kernels might not import successfully.
- Typically, you import a compressed version of a disk image; the expanded disk image cannot exceed 1 TiB.
- Make sure that you have at least 250 MB of available disk space for installing drivers and other software on any VM you want to import into an Amazon EC2 AMI running Microsoft Windows or Linux.
- Multiple network interfaces are not currently supported. When converted and imported, your instance will have a single virtual NIC using DHCP for address assignment.
- Internet Protocol version 6 (IPv6) IP addresses are not supported.
- For vCenter 4.0 and vSphere 4.0 users, remove any attached CD-ROM images or ISOs from the virtual machine.
- VMs that are created as the result of a P2V conversion are not supported by Amazon EC2 VM import. A P2V conversion occurs when a disk image is created by performing a Linux or Windows installation process on a physical machine and then importing a copy of that Linux or Windows installation into a VM.
- Amazon VM Import does not install the single root I/O virtualization (SR-IOV) drivers except for imports of Microsoft Windows Server 2012 R2 VMs. These drivers are not required unless you plan to use enhanced networking, which provides higher performance (packets per second), lower latency, and lower jitter. To enable enhanced networking on a c3 or i2 instance type after you import your VM, see Enabling Enhanced Networking on Windows Instances in a VPC (p. 510). For Microsoft Windows Server 2012 R2 VMs, SR-IOV driver are automatically installed as a part of the import process.
- In connection with your use of your own Microsoft licenses, such as through MSDN or Windows Software Assurance Per User, to run Microsoft Software on AWS through a bring your own license (BYOL) model:
  1. Your BYOL instances will be priced at the prevailing Amazon EC2 Linux instance pricing (set out at Amazon EC2 Instance Purchasing Options), provided that you (a) run on a Dedicated Instance (For more information, see Dedicated Instances); (b) launch from VMs sourced from software binaries provided by you using VM Import/Export, which will be subject to the then-current terms and abilities of VM Import/Export; (c) designate the instances as BYOL instances (i.e., declare the appropriate platform type flag in the services); (d) run the instances within your designated AWS regions, and where AWS offers the BYOL model; and (e) activate using Microsoft keys that you provide or are used in your Key Management System.
  2. You must account for the fact that when you start an Amazon EC2 instance, it can run on any one of many servers within an Availability Zone. This means that each time you start an Amazon EC2 instance (including a stop/start), it may run on a different server within an Availability Zone. You must account for this fact in light of the limitations on license reassignment as described in the Microsoft Volume Licensing Product Use Rights (PUR)/Product Terms (PT) available at Volume Licensing for Microsoft Products and Online Services, or consult your specific use rights to determine if your rights are consistent with this usage.
  3. You must be eligible to use the BYOL program for the applicable Microsoft software under your agreement(s) with Microsoft, for example, under your MSDN user rights or under your Windows Software Assurance Per User Rights. You are solely responsible for obtaining all required licenses and for complying with all applicable Microsoft licensing requirements, including the PUR/PT. Further, you must have accepted Microsoft's End User License Agreement (Microsoft EULA), and by using the Microsoft Software under the BYOL program, you agree to the Microsoft EULA.

4. AWS recommends that you consult with your own legal and other advisers to understand and comply with the applicable Microsoft licensing requirements. Usage of the Services (including usage of the **licenseType** parameter and **BYOL** flag) in violation of your agreement(s) with Microsoft is not authorized or permitted.

Known Limitations for Importing a VM into Amazon EC2 Using ImportInstance

Importing instances and volumes is subject to the following limitations:

- You can have up to five import tasks per region in progress at the same time. To request an increase to this limit, contact AWS Support. Tasks must complete within 7 days of the start date.
- Imported instances create EC2 instances that use Hardware Virtual Machine (HVM) virtualization. Creating instances that use Paravirtual (PV) virtualization using VM Import is not supported. Linux PVHVM drivers are supported within imported instances.
- Imported Red Hat Enterprise Linux (RHEL) instances must use Cloud Access (BYOL) licenses.
- Imported Linux instances must use 64-bit images. Importing 32-bit Linux images is not supported.
- Imported Linux instances should use default kernels for best results. VMs that use custom Linux kernels might not import successfully.
- Typically, you import a compressed version of a disk image; the expanded disk image cannot exceed 1 TiB.
- Make sure your VM only uses a single disk. Importing a VM with more than one disk is not supported. For Linux VMs, /boot and / can be located in different partitions, but they need to be on the same disk.

  We suggest that you import the VM with only the boot volume, and import any additional disks using the ec2-import-volume command. After the `ImportInstance` task is complete, use the ec2-attach-volume command to associate the additional volumes with your instance.
- Virtual Hard Disk (VHD) images must be dynamic.
- Make sure that you have at least 250 MB of available disk space for installing drivers and other software on any VM you want to import into an Amazon EC2 instance running Microsoft Windows or Linux.
- Imported instances automatically have access to the Amazon EC2 instance store, which is temporary disk storage located on disks that are physically attached to the host computer. You cannot disable this during import. For more information about instance storage, see Amazon EC2 Instance Store (p. 577).
- Multiple network interfaces are not currently supported. When converted and imported, your instance will have a single virtual NIC using DHCP for address assignment.
- Internet Protocol version 6 (IPv6) IP addresses are not supported.
- For vCenter 4.0 and vSphere 4.0 users, remove any attached CD-ROM images or ISOs from the virtual machine.
- Amazon VM Import does not install the single root I/O virtualization (SR-IOV) drivers on the c3 and i2 instance types, except for imports of Microsoft Windows Server 2012 R2 VMs. These drivers are not required unless you plan to use enhanced networking, which provides higher performance (packets per second), lower latency, and lower jitter. To enable enhanced networking on a c3 or i2 instance type after you import your VM, see Enabling Enhanced Networking on Windows Instances in a VPC (p. 510). For Microsoft Windows Server 2012 R2 VMs, SR-IOV driver are automatically installed as a part of the import process.
- You cannot import Microsoft Windows instances that use the bring your own license (BYOL) model. To import these instance types, see Importing a VM into Amazon EC2 Using ImportImage (p. 177).

Known Limitations for Exporting a VM from Amazon EC2

Exporting instances and volumes is subject to the following limitations:

- You can have up to five export tasks per region in progress at the same time.
- You cannot export Amazon Elastic Block Store (Amazon EBS) data volumes.

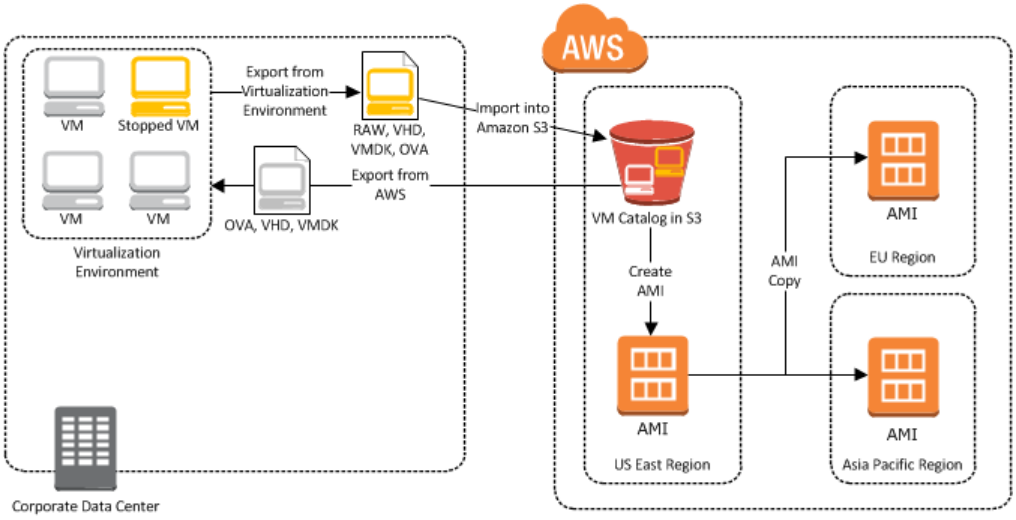
- You cannot export an instance or AMI that has more than one virtual disk.
- You cannot export an instance or AMI that has more than one network interface.
- You cannot export an instance or AMI from Amazon EC2 unless you previously imported it into Amazon EC2 from another virtualization environment.
- You cannot export an instance or AMI from Amazon EC2 if you've shared it from another AWS account.

Importing a VM into Amazon EC2 Using ImportImage

You can import a virtual machine (VM) from your virtualization environment such as Citrix Xen, Microsoft Hyper-V, or VMware vSphere, and import it as an AMI in Amazon EC2. For more information about how to launch an Amazon EC2 instance from an AMI, see Launch Your Instance.

To use your VM in Amazon EC2, you must first export it from the virtualization environment, and then import it into Amazon EC2 using the AWS Command Line Interface (AWS CLI) or API tools.

The following diagram shows the process of exporting a VM from your on-premises virtualization environment to AWS.



Whether you use the CLI or the API, you will follow the same steps for importing VMs or volumes into Amazon EC2. This is the process for using the CLI.

To import a VM into Amazon EC2

1. Install the AWS CLI. For more information, see Step 1: Install the AWS CLI (p. 178).
2. Prepare the VM for import to Amazon EC2. For more information, see Step 2: Prepare Your VM (p. 178).
3. Export the VM from the virtualization environment. For more information, see Step 3: Export Your VM from Its Virtual Environment (p. 180).
4. Import the VM into Amazon EC2. For information, see Step 4: Importing Your VM into Amazon EC2 (p. 180).
5. Launch the instance in Amazon EC2. For more information, see Step 5: Launch the instance in Amazon EC2 (p. 185).

Step 1: Install the AWS CLI

You can install the AWS CLI to import your Citrix, Microsoft Hyper-V, or VMware vSphere virtual machines into Amazon EC2. For more information about installing the AWS CLI, see Configuring the AWS Command Line Interface in the *AWS Command Line Interface User Guide*.

You'll use the following commands in the AWS CLI to import VMs in the supported formats:

Command	Description
import-image	Creates a new import image task using metadata from the specified disk image(s) and creates an Amazon Machine Image (AMI).
import-snapshot	Creates a new import snapshot task using metadata from the specified disk image and imports the snapshot into Amazon EBS.
describe-import-image-tasks	Lists and describes your import tasks.
describe-import-snapshot-tasks	Lists and describes your snapshot import tasks.
cancel-import-task	Cancels an active import task.

Step 2: Prepare Your VM

Use the following guidelines to configure your VM before exporting it from the virtualization environment.

- Review the prerequisites. For more information, see VM Import/Export Prerequisites (p. 169).
- Disable any antivirus or intrusion detection software on your VM. These services can be re-enabled after the import process is complete.
- Uninstall the VMware Tools from your VMware VM.
- Disconnect any CD-ROM drives (virtual or physical).
- Set your network to DHCP instead of a static IP address. If you want to assign a static private IP address, be sure to use a non-reserved private IP address in your VPC subnet. Amazon Virtual Private Cloud (Amazon VPC) reserves the first four private IP addresses in a VPC subnet.
- Shut down your VM before exporting it from your virtualization environment.

Windows

- Enable Remote Desktop (RDP) for remote access.
- Make sure that your host firewall (Windows firewall or similar), if configured, allows access to RDP. Otherwise, you will not be able to access your instance after the import is complete.
- Make sure that the administrator account and all other user accounts use secure passwords. All accounts must have passwords or the importation might fail.
- Make sure that your Windows VM has .NET Framework 3.5 or later installed, as required by Amazon Windows EC2Config Service.
- You can run System Preparation (Sysprep) on your Windows Server 2008 or Windows Server 2012 VM images before or after they are imported. If you run Sysprep before importing your VM, the importation process adds an answer file (unattend.xml) to the VM that automatically accepts the End User License Agreement (EULA) and sets the locale to EN-US. If you choose to run Sysprep after importation, we recommend that you use the Amazon EC2 Config service to run Sysprep.

  To include your own answer file instead of the default (unattend.xml):

1. Copy the sample unattend.xml file below and set the **processorArchitecture** parameter to **x86** or **amd64**, depending on your OS architecture:

```
<?xml version='1.0' encoding='UTF-8'?>
<unattend xmlns:wcm='http://schemas.microsoft.com/WMIConfig/2002/State' xm
lns='urn:schemas-microsoft-com:unattend'>
 <settings pass='oobeSystem'>
  <component versionScope='nonSxS' processorArchitecture='x86 or amd64'
name='Microsoft-Windows-International-Core' publicKeyToken='31bf3856ad364e35'
 language='neutral'>
   <InputLocale>en-US</InputLocale>
   <SystemLocale>en-US</SystemLocale>
   <UILanguage>en-US</UILanguage>
   <UserLocale>en-US</UserLocale>
  </component>
  <component versionScope='nonSxS' processorArchitecture='x86 or amd64'
name='Microsoft-Windows-Shell-Setup' publicKeyToken='31bf3856ad364e35' lan
guage='neutral'>
   <OOBE>
    <HideEULAPage>true</HideEULAPage>
    <SkipMachineOOBE>true</SkipMachineOOBE>
    <SkipUserOOBE>true</SkipUserOOBE>
   </OOBE>
  </component>
 </settings>
</unattend>
```

2. Save the file in the **C:\Windows\Panther** directory with the name **unattend.xml**.
3. Run Sysprep with the **/oobe** and **/generalize** options.
4. Shutdown the VM and export it from your virtualization environment.

- Disable Autologon on your Windows VM.
- Open **Control Panel** > **System and Security** > **Windows Update**. In the left pane, choose **Change settings**. Choose the desired setting. Be aware that if you choose **Download updates but let me choose whether to install them** (the default value) the update check can temporarily consume between 50% and 99% of CPU resources on the instance. The check usually occurs several minutes after the instance starts. Make sure that there are no pending Microsoft updates, and that the computer is not set to install software when it reboots.
- Apply the following hotfixes:
  - You cannot change system time if RealTimeIsUniversal registry entry is enabled in Windows
  - High CPU usage during DST changeover in Windows Server 2008, Windows 7, or Windows Server 2008 R2
- Enable the RealTimeIsUniversal registry. For more information, see Setting the Time for a Windows Instance (p. 281).

Linux

- Enable Secure Shell (SSH) for remote access.
- Make sure that your host firewall (such as Linux iptables) allows access to SSH. Otherwise, you will not be able to access your instance after the import is complete.
- Make sure that you have configured a non-root user to use public key-based SSH to access your instance after it is imported. The use of password-based SSH and root login over SSH are both possible, but not recommended. The use of public keys and a non-root user is recommended because it is more secure. VM Import will not configure an *ec2-user* account as part of the import process.
- Make sure that your Linux VM uses GRUB (GRUB legacy) or GRUB 2 as its bootloader.

- Make sure that your Linux VM uses a root filesystem is one of the following: EXT2, EXT3, EXT4, Btrfs, JFS, or XFS.

Step 3: Export Your VM from Its Virtual Environment

After you have prepared your VM for export, you can export it from your virtualization environment. For information about how to export a VM from your virtualization environment, see the documentation for Citrix, Microsoft Hyper-V, or VMware vCenter virtualization environment.

Citrix: For more information, see Export VMs as OVF/OVA at the Citrix website.

Microsoft Hyper-V: For more information, see Hyper-V - Export & Import at the Microsoft website.

VMware: For more information, see Export an OVF Template at the VMware website.

Step 4: Importing Your VM into Amazon EC2

After exporting your VM from your virtualization environment, you can import it into Amazon EC2. The import process is the same regardless of the origin of the VM.

Here are some important things to know about your VM instance, as well as some security and storage recommendations:

- Amazon EC2 automatically assigns a private DHCP IP address to your instance. The DNS name and IP address are available through the `ec2-describe-instances` command when the instance starts running. If the instance is imported into a VPC, it will not get a public IP address, though the subnet has auto-assign public IP enabled, for security reasons. However, you may create an Elastic IP address (EIP) and attach it to the imported instance.
- Your instance will have only one Ethernet network interface.
- We recommend that your Windows instances contain strong passwords for all user accounts. We recommend that your Linux instances use public keys for SSH.
- For Windows instances, we recommend that you install the latest version of the Amazon Windows EC2Config Service after you import your virtual machine into Amazon EC2.

To import a VM in OVA format into Amazon EC2

You can upload your VMs in OVA format to your Amazon S3 bucket using the upload tool of your choice. After you upload your VM to Amazon S3, you can use the AWS CLI to import your OVA image. These tools accept either a URL (public Amazon S3 file, a signed GET URL for private Amazon S3 files) or the Amazon S3 bucket and path to the disk file.

Use `aws ec2 import-image` to create a new import instance task.

The syntax of the command is as follows:

```
C:\> aws ec2 import-image --cli-input-json "{  \"Description\": \"Windows 2008
 OVA\", \"DiskContainers\": [ { \"Description\": \"First CLI task\", \"UserBuck
et\": { \"S3Bucket\": \"my-import-bucket\", \"S3Key\" : \"my-windows-2008-
vm.ova\" } } ]}"
```

Example response

```
<ImportImageResponse xmlns="http://ec2.amazonaws.com/doc/2015-03-01/">
    <progress>2</progress>
```

```
    <importTaskId>import-ami-fgxn195v</importTaskId>
    <status>active</status>
    <description>Windows 2008 OVA</description>
    <snapshotTaskDetailSet>
        <item>
            <diskImageSize>0.0</diskImageSize>
            <userBucket>
                <s3Bucket>my-import-bucket</s3Bucket>
                <s3Key>my-windows-2008-vm.ova</s3Key>
            </userBucket>
        </item>
    </snapshotTaskDetailSet>
    <licenseType>AWS</licenseType>
    <statusMessage>pending</statusMessage>
    <requestId>1571e127-d6d8-4984-b4f1-3a21e9dbdcb5</requestId>
</ImportImageResponse>
```

To import a VM with multiple explicit disks into Amazon EC2

After you upload your VM disk images to Amazon S3, you can use the AWS CLI to import your disk images or snapshots. These tools accept either a URL (public Amazon S3 file, a signed GET URL for private Amazon S3 files) or the Amazon S3 bucket and path to the disk file. You can also use Amazon EBS snapshots as input to the ImportImage API.

Example using the aws ec2 import-image command with multiple explicit disks

Use **aws ec2 import-image** command to a new import instance task.

```
C:\> aws ec2 import-image --cli-input-json "{  \"Description\": \"Windows 2008
 VMDKs\", \"DiskContainers\": [ { \"Description\": \"Second CLI task\",
\"UserBucket\": { \"S3Bucket\": \"my-import-bucket\", \"S3Key\" : \"my-windows-
2008-vm-disk1.vmdk\" } }, { \"Description\": \"First CLI task\", \"UserBucket\":
 { \"S3Bucket\": \"my-import-bucket\", \"S3Key\" : \"my-windows-2008-vm-
disk2.vmdk\" } } ] }"
```

Example response

```
<ImportImageResponse xmlns="http://ec2.amazonaws.com/doc/2015-03-01/">
    <progress>2</progress>
    <importTaskId>import-ami-fgxn591c</importTaskId>
    <status>active</status>
    <description>Windows 2008 VMDKs</description>
    <snapshotTaskDetailSet>
        <item>
            <diskImageSize>0.0</diskImageSize>
            <userBucket>
                <s3Bucket>my-import-bucket</s3Bucket>
                <s3Key>my-windows-2008-vm-disk1.vmdk</s3Key>
            </userBucket>
        </item>
        <item>
            <diskImageSize>0.0</diskImageSize>
            <userBucket>
                <s3Bucket>my-import-bucket</s3Bucket>
                <s3Key>my-windows-2008-vm-disk2.vmdk</s3Key>
```

```
            </userBucket>
        </item>
    </snapshotTaskDetailSet>
    <licenseType>AWS</licenseType>
    <statusMessage>pending</statusMessage>
    <requestId>1571e127-d6d8-4984-b4f1-3a21e9dbdcb5</requestId>
</ImportImageResponse>
```

Checking on the Status of Your Import Image Task

The `aws ec2 describe-import-image-tasks` command returns the status of an import task. Status values include the following:

- **active—**Your task is active and currently in progress.
- **deleting—**Your task is currently being cancelled.
- **deleted—**Your task is canceled.
- **completed—**Your task is complete and the AMI is ready to use.

To check the status of your import task

Use the `aws ec2 describe-import-image-tasks` command to return the status of the task. The syntax of the command is as follows:

Example using the aws ec2 describe-import-image-tasks command:

The following example enables you to see the status of your import task.

```
C:\> aws ec2 describe-import-image-tasks --cli-input-json "{ \"ImportTaskIds\":
 [\"import-ami-fgxn195v\"], \"NextToken\": \"abc\", \"MaxResults\": 10 } "
```

Example Response

The following response shows the output from the `aws ec2 describe-import-image-tasks` command.

```
<DescribeImportImageTasksResponse xmlns="http://ec2.amazonaws.com/doc/2015-03-
01/">
    <importImageTaskSet>
        <item>
            <platform>Windows</platform>
            <importTaskId>import-ami-fgs8im0c</importTaskId>
            <imageId>ami-4a6c2722</imageId>
            <status>completed</status>
            <description>Linux OVA</description>
            <architecture>x86_64</architecture>
            <snapshotTaskDetailSet>
                <item>
                    <diskImageSize>3.115815424E9</diskImageSize>
                    <deviceName>/dev/sda1</deviceName>
                    <description>First CLI task</description>
                    <format>VMDK</format>
                    <url>https://mys3bucket/vms/my-linux-vm.ova?AWSAccessKey
Id=myAccessKeyId&Expires=expirationDate&Signature=mySignature</url>
                </item>
            </snapshotTaskDetailSet>
            <licenseType>AWS</licenseType>
```

```
        </item>
    </importImageTaskSet>
    <requestId>377ec1ca-6a47-42f5-8b84-aa07ff87f7b0</requestId>
</DescribeImportImageTasksResponse>
```

Importing Your Disk Images into Amazon EBS

This section describes how to import your disks into Amazon EBS snapshots, and then create Amazon EBS volumes later. Amazon EC2 supports importing RAW, Virtual Hard Disk (VHD), and ESX Virtual Machine Disk (VMDK) disk formats.

After you have exported your virtual machine from the virtualization environment, importing the volume to Amazon EBS is a single-step process. You create an upload task to upload the disk image to Amazon S3 and then create an import task to use the volume.

To import a disk image into Amazon EBS

1. Use the aws ec2 import-snapshot command to upload your volume into Amazon EBS.

   Example using the aws ec2 import-snapshot command.

   ```
   C:\> aws ec2 import-snapshot --cli-input-json "{  \"Description\": \"Windows
    2008 VMDK\", \"DiskContainer\": { \"Description\": \"First CLI snap\",
   \"Url\": \"https://mys3bucket/vms/Win_2008_Server_Enterprise_R2_64-bit.vm
   dk?AWSAccessKeyId=myaccesskey&Expires=expirationdate&Signature=signature\"
    }, \"ClientToken\": \"abc\" }"
   ```

   Example response

   ```
   <ImportSnapshotResponse xmlns="http://ec2.amazonaws.com/doc/2015-03-01/">
       <snapshotTaskDetail>
           <diskImageSize>0.0</diskImageSize>
           <progress>3</progress>
           <status>active</status>
           <description>Windows 2008 VMDK</description>
           <url>https://mys3bucket/vms/Win_2008_Server_Enterprise_R2_64-
   bit.vmdk?AWSAccessKeyId=myaccesskey&Expires=expirationdate&Signature=signa
   ture\</url>
           <statusMessage>pending</statusMessage>
       </snapshotTaskDetail>
       <importTaskId>import-snap-ffy5pvea</importTaskId>
       <description>Windows 2008 VMDK</description>
       <requestId>2ef5652d-6816-4c20-89b2-a4bbb0560190</requestId>
   </ImportSnapshotResponse>
   ```

2. Use the `aws ec2 describe-import-snapshot-tasks` command to confirm that your snapshot imported successfully.

   Example using the aws ec2 describe-import-snapshot-tasks command

   ```
   C:\> aws ec2 describe-import-snapshot-tasks --cli-input-json "{ \"Import
   TaskIds\": [\"import-snap-fgr1mmg7\"], \"NextToken\": \"abc\", \"MaxResults\":
    10 } "
   ```

   Example response

```
<DescribeImportSnapshotTasksResponse xmlns="http://ec2.amazonaws.com/doc/2015-03-01/">
    <importSnapshotTaskSet>
        <item>
            <snapshotTaskDetail>
                <diskImageSize>3.115815424E9</diskImageSize>
                <progress>22</progress>
                <status>active</status>
                <description>Windows 2008 VMDK</description>
                <format>VMDK</format>
              <url>https://mys3bucket/vms/Win_2008_Server_Enterprise_R2_64-bit.vmdk?AWSAccessKeyId=myaccesskey&Expires=expirationdate&Signature=signature\</url>
                <statusMessage>validated</statusMessage>
            </snapshotTaskDetail>
            <importTaskId>import-snap-fgr1mmg7</importTaskId>
            <description>Windows 2008 VMDK</description>
        </item>
    </importSnapshotTaskSet>
    <requestId>3ec7adc5-001a-454f-abc3-820c8a91c353</requestId>
</DescribeImportSnapshotTasksResponse>
```

The status in this example is **active**, which means the import is still ongoing.

3. Use `aws ec2 create-volume` to create a volume from the Amazon EBS snapshot. The following example creates a volume from a snapshot. Make sure you select an availability zone where the instance resides so that the Amazon EBS volume can be attached to the Amazon EC2 instance.

```
C:\> aws ec2 create-volume --availability-zone us-east-1a –snapshot-id snap-abcd1234
```

Example output

```
{
    "AvailabilityZone": "us-east-1a",
    "VolumeId": "vol-1234abcd",
    "State": "creating",
    "SnapshotId": "snap-abcd1234"
}
```

4. Use `aws ec2 attach-volume` to attach the Amazon EBS volume to one of your existing Amazon EC2 instances. The following example attaches the volume, vol-1234abcd, to the i-abcd1234 instance on the device, /dev/sdf.

```
C:\> aws ec2 attach-volume --volume-id vol-1234abcd --instance-id i-abcd1234
 --device /dev/sdf
```

Example output

```
{
    "AttachTime": "YYYY-MM-DDTHH:MM:SS.000Z",
    "InstanceId": "i-abcd1234",
    "VolumeId": "vol-1234abcd",
    "State": "attaching",
```

```
    "Device": "/dev/sdf"
}
```

Canceling an Import Task

Use the `aws ec2 cancel-import-task` command to cancel an active import task. The task can be the import of an AMI or snapshot.

To cancel an import task

Use the task ID of the import you want to cancel with the `aws ec2 cancel-import-task` command.

Example using the aws ec2 cancel-import-task command

The following example cancels the upload associated with the task ID `import-ami-fg4z7c9h`.

```
C:\> aws ec2 cancel-import-task --import-task-id "import-ami-fg4z7c9h"
```

Example response

```
<CancelImportTaskResponse xmlns="http://ec2.amazonaws.com/doc/2015-03-01/">
    <importTaskId>import-ami-fg4z7c9h</importTaskId>
    <state>active</state>
    <previousState>deleting</previousState>
    <requestId>1e5abd4c-b8de-4b3c-8c1a-73d93b006c1f</requestId>
</CancelImportTaskResponse>
```

Step 5: Launch the instance in Amazon EC2

After the `aws ec2 import-image` task is complete, you will see your AMI in the Amazon EC2 console. You can select this AMI and then launch an Amazon EC2 instance based on this AMI.

To launch an Amazon EC2 instance based on your AMI

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see Regions and Endpoints.
3. In the navigation pane, click **AMIs**.
4. In the content pane, select the AMI, and then click **Launch**.

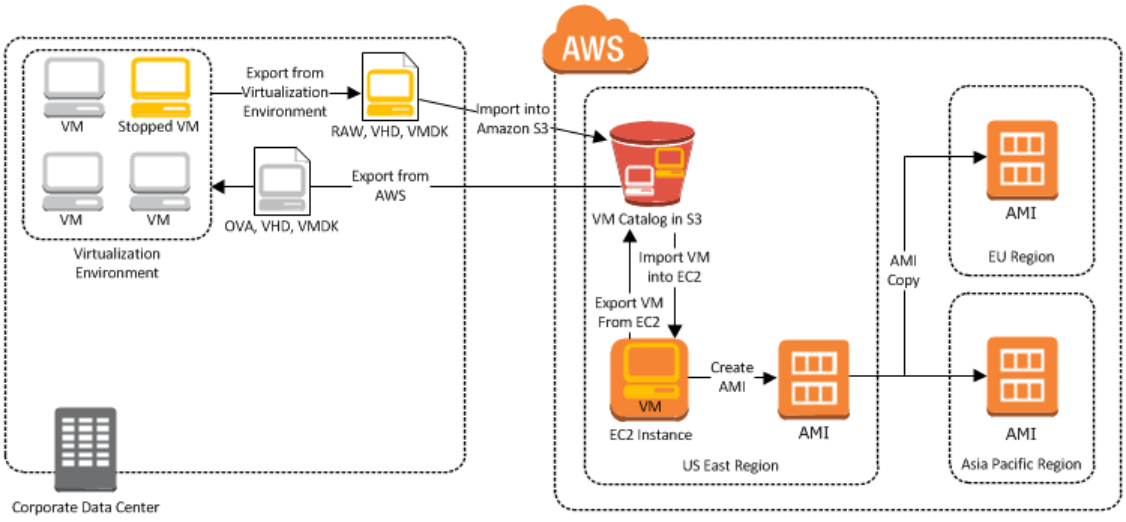
Importing a VM into Amazon EC2 Using ImportInstance

There are two ways you can launch an instance in Amazon EC2. You can launch an instance from an Amazon Machine Image (AMI), or, you can launch an instance from a virtual machine (VM) that you imported from a virtualization environment such as Citrix Xen, Microsoft Hyper-V, or VMware vSphere. This section covers importing a VM and launching it as an Amazon EC2 instance. This method only supports single-volume VMs. To import VMs with multiple volumes, see Importing a VM into Amazon EC2 Using ImportImage (p. 177). For more information about how to launch an Amazon EC2 instance from an AMI, see Launch Your Instance (p. 206).

To use your VM as an instance in Amazon EC2, you must first export it from the virtualization environment, and then import it to Amazon EC2 using the Amazon EC2 command line interface (CLI) or API tools. If you're importing a VM from VMware vCenter, you can also use the AWS Connector for vCenter to export a VM from VMware and import it into Amazon EC2. For more information, see Migrating Your Virtual Machine to Amazon EC2 Using AWS Connector for vCenter in the *AWS Management Portal for vCenter User Guide*.

Important
You cannot use ImportInstance to import Microsoft Windows instances that use the bring your own license (BYOL) model. To import these instance types, see Importing a VM into Amazon EC2 Using ImportImage (p. 177).

The following diagram shows the process of exporting a VM from your on-premises virtualization environment to AWS.



Whether you use the CLI or the API, you will follow the same steps for importing VMs or volumes into Amazon EC2. This is the process for using the CLI.

To import a VM into Amazon EC2

1. Install the CLI. For more information, see Step 1: Install the Amazon EC2 CLI (p. 186).
2. Prepare the VM for import to Amazon EC2. For more information, see Step 2: Prepare Your VM (p. 187).
3. Export the VM from the virtualization environment. For more information, see Step 3: Export Your VM from Its Virtual Environment (p. 189).
4. Import the VM into Amazon EC2. For information, see Step 4: Importing Your VM into Amazon EC2 (p. 189).
5. Launch the instance in Amazon EC2. For more information, see Step 5: Launch the instance in Amazon EC2 (p. 195).

Step 1: Install the Amazon EC2 CLI

You need to install the Amazon EC2 CLI to import your Citrix, Microsoft Hyper-V, or VMware vSphere virtual machines into Amazon EC2 or to export them from Amazon EC2. If you haven't already installed the Amazon EC2 CLI, see Setting Up the Amazon EC2 Tools.

You'll use the following Amazon EC2 commands to import or export a VM.

Command	Description
ec2-import-instance	Creates a new import instance task using metadata from the specified disk image and imports the instance to Amazon EC2.
ec2-import-volume	Creates a new import volume task using metadata from the specified disk image and imports the volume to Amazon EC2.
ec2-resume-import	Resumes the upload of a disk image associated with an import instance or import volume task ID.
ec2-describe-conversion-tasks	Lists and describes your import tasks.
ec2-cancel-conversion-task	Cancels an active import task. The task can be the import of an instance or volume.
ec2-delete-disk-image	Deletes a partially or fully uploaded disk image for import from an Amazon S3 bucket.
ec2-create-image-export-task	Exports a running or stopped instance to an Amazon S3 bucket.
ec2-cancel-export-task	Cancels an active export task.
ec2-describe-export-tasks	Lists and describes your export tasks, including the most recent canceled and completed tasks.

For information about these commands and other Amazon EC2 commands, see the Amazon EC2 Command Line Reference.

Step 2: Prepare Your VM

Use the following guidelines to configure your VM before exporting it from the virtualization environment.

- Review the prerequisites. For more information, see VM Import/Export Prerequisites (p. 169).
- Disable any antivirus or intrusion detection software on your VM. These services can be re-enabled after the import process is complete.
- Uninstall the VMware Tools from your VMware VM.
- Disconnect any CD-ROM drives (virtual or physical).
- Set your network to DHCP instead of a static IP address. If you want to assign a static private IP address, be sure to use a non-reserved private IP address in your VPC subnet. Amazon Virtual Private Cloud (Amazon VPC) reserves the first four private IP addresses in a VPC subnet.
- Shut down your VM before exporting it.

Windows

- Enable Remote Desktop (RDP) for remote access.
- Make sure that your host firewall (Windows firewall or similar), if configured, allows access to RDP. Otherwise, you will not be able to access your instance after the import is complete.
- Make sure that the administrator account and all other user accounts use secure passwords. All accounts must have passwords or the importation might fail.
- Make sure that your Windows VM has .NET Framework 3.5 or later installed, as required by Amazon Windows EC2Config Service.

- You can run System Preparation (Sysprep) on your Windows Server 2008 or Windows Server 2012 VM images before or after they are imported. If you run Sysprep before importing your VM, the importation process adds an answer file (unattend.xml) to the VM that automatically accepts the End User License Agreement (EULA) and sets the locale to EN-US. If you choose to run Sysprep after importation, we recommend that you use the Amazon EC2 Config service to run Sysprep.

  To include your own answer file instead of the default (unattend.xml):

  1. Copy the sample unattend.xml file below and set the **processorArchitecture** parameter to **x86** or **amd64**, depending on your OS architecture:

```
<?xml version='1.0' encoding='UTF-8'?>
<unattend xmlns:wcm='http://schemas.microsoft.com/WMIConfig/2002/State' xm
lns='urn:schemas-microsoft-com:unattend'>
 <settings pass='oobeSystem'>
  <component versionScope='nonSxS' processorArchitecture='x86 or amd64'
name='Microsoft-Windows-International-Core' publicKeyToken='31bf3856ad364e35'
 language='neutral'>
   <InputLocale>en-US</InputLocale>
   <SystemLocale>en-US</SystemLocale>
   <UILanguage>en-US</UILanguage>
   <UserLocale>en-US</UserLocale>
  </component>
  <component versionScope='nonSxS' processorArchitecture='x86 or amd64'
name='Microsoft-Windows-Shell-Setup' publicKeyToken='31bf3856ad364e35' lan
guage='neutral'>
   <OOBE>
    <HideEULAPage>true</HideEULAPage>
    <SkipMachineOOBE>true</SkipMachineOOBE>
    <SkipUserOOBE>true</SkipUserOOBE>
   </OOBE>
  </component>
 </settings>
</unattend>
```

  2. Save the file in the **C:\Windows\Panther** directory with the name **unattend.xml**.
  3. Run Sysprep with the **/oobe** and **/generalize** options.
  4. Shutdown the VM and export it from your virtualization environment.
- Disable Autologon on your Windows VM.
- Make sure that there are no pending Microsoft updates, and that the computer is not set to install software when it reboots.
- Apply the following hotfixes:
  - You cannot change system time if RealTimeIsUniversal registry entry is enabled in Windows
  - High CPU usage during DST changeover in Windows Server 2008, Windows 7, or Windows Server 2008 R2
- Enable the RealTimeIsUniversal registry. For more information, see Setting the Time for a Windows Instance (p. 281).

Linux

- Enable Secure Shell (SSH) for remote access.
- Make sure that your host firewall (such as Linux iptables) allows access to SSH. Otherwise, you will not be able to access your instance after the import is complete.
- Make sure that you have configured a non-root user to use public key-based SSH to access your instance after it is imported. The use of password-based SSH and root login over SSH are both possible,

but not recommended. The use of public keys and a non-root user is recommended because it is more secure. VM Import will not configure an *ec2-user* account as part of the import process.

- Make sure that your Linux VM uses GRUB (GRUB legacy) or GRUB 2 as its bootloader.
- Make sure that your Linux VM uses a root filesystem is one of the following: EXT2, EXT3, EXT4, Btrfs, JFS, or XFS.

Step 3: Export Your VM from Its Virtual Environment

After you have prepared your VM for export, you can export it from your virtualization environment. For information about how to export a VM from your virtualization environment, see the documentation for Citrix, Microsoft Hyper-V, or VMware vCenter virtualization environment.

Citrix: For more information, see Export VMs as OVF/OVA at the Citrix website.

Microsoft Hyper-V: For more information, see Hyper-V - Export & Import at the Microsoft website.

VMware: For more information, see Export an OVF Template at the VMware website.

Step 4: Importing Your VM into Amazon EC2

After exporting your VM from your virtualization environment, you can import it into Amazon EC2. The import process is the same regardless of the origin of the VM.

Here are some important things to know about your VM instance, as well as some security and storage recommendations:

- Amazon EC2 automatically assigns a private DHCP IP address to your instance. The DNS name and IP address are available through the `ec2-describe-instances` command when the instance starts running. If the instance is imported into a VPC, it will not get a public IP address, though the subnet has auto-assign public IP enabled, for security reasons. However, you may create an Elastic IP address (EIP) and attach it to the imported instance.
- Your instance has only one Ethernet network interface.
- To specify a subnet to use when you create the import task, use the `--subnet` *`subnet_id`* option with the `ec2-import-instance` command; otherwise, your instance will use a public IP address. We recommend that you use a restrictive security group to control access to your instance.
- We recommend that your Windows instances contain strong passwords for all user accounts. We recommend that your Linux instances use public keys for SSH.
- For Windows instances, we recommend that you install the latest version of the Amazon Windows EC2Config Service after you import your virtual machine into Amazon EC2.

To import a VM into Amazon EC2

Use ec2-import-instance to create a new import instance task.

The syntax of the command is as follows:

```
ec2-import-instance disk_image_filename -f file_format -t instance_type -a ar
chitecture -b s3_bucket_name -o owner -w secret_key -p platform_name
```

If the import of the VM is interrupted, you can use the `ec2-resume-import` command to resume the import from where it stopped. For more information, see Resuming an Upload (p. 193).

Example (Windows)

The following command creates an import instance task that imports a Windows Server 2008 SP2 (32-bit) VM.

```
C:\> ec2-import-instance ./WinSvr8-2-32-disk1.vmdk  –f VMDK -t m1.small -a i386
 -b myawsbucket -o AKIAIOSFODNN7EXAMPLE -w wJalrXUtnFEMI/K7MDENG/bPxRfi
CYEXAMPLEKEY -p Windows
```

This request uses the VMDK file, `WinSvr8-2-32-disk1.vmdk`, to create the import task. (Note that you can alternatively use VHD or RAW format.) If you do not specify a size for the requesting volume using the `-s` parameter, a volume size based on the disk image file is used. The output is similar to the following.

```
Requesting volume size: 25 GB
Disk image format: Stream-optimized VMDK
Converted volume size: 26843545600 bytes (25.00 GiB)
Requested EBS volume size: 26843545600 bytes (25.00 GiB)
TaskType        IMPORTINSTANCE  TaskId  import-i-fhbx6hua       ExpirationTime
  2011-09-09T15:03:38+00:00       Status  active  StatusMessage   Pending In
stanceID      i-6ced060c
DISKIMAGE       DiskImageFormat VMDK    DiskImageSize   5070303744
VolumeSize      25      AvailabilityZone        us-east-1c      Approximate
BytesConverted       0       Status  active StatusMessage    Pending
Creating new manifest at testImport/9cba4345-b73e-4469-8106-
2756a9f5a077/Win_2008_R1_EE_64.vmdkmanifest.xml
Uploading the manifest file
Uploading 5070303744 bytes across 484 parts
0% |----------------------------------------------------| 100%
   |====================================================|
Done
```

Example (Linux)

The following example creates an import instance task that imports a 64-bit Linux VM.

```
$ ec2-import-instance rhel6.4-64bit-disk.vhd -f vhd -t m3.xlarge -a x86_64 -b
myawsbucket -o AKIAIOSFODNN7EXAMPLE –w wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
 -p Linux
```

This request uses the VHD file, **rhel6.4-64bit-disk.vhd**, to create the import task. The output is similar to the following.

```
Requesting volume size: 8 GB
TaskType        IMPORTINSTANCE  TaskId  import-i-ffnzq636       ExpirationTime
  2013-12-12T22:55:18Z    Status      active  StatusMessage   Pending InstanceID
      i-a56ab6dd
DISKIMAGE       DiskImageFormat VHD    DiskImageSize   861055488
VolumeSize     8      AvailabilityZone    us-east-1d     ApproximateBytesCon
verted       0       Status  active  StatusMessage   Pending
Creating new manifest at myawsbucket/b73bae14-7ec5-4122-8958-
4234028e1d9f/rhel6.4-64bit-disk.vhdmanifest.xml
Uploading the manifest file
Uploading 861055488 bytes across 83 parts
0% |----------------------------------------------------| 100%
   |====================================================|
```

```
Done

Average speed was 11.054 MBps

The disk image for import-i-ffnzq636 has been uploaded to Amazon S3 where it
is being converted into
an EC2 instance. You may monitor the progress of this task by running ec2-de
scribe-conversion-tasks.
When the task is completed, you may use ec2-delete-disk-image to remove the
image from S3.
```

Checking on the Status of Your Import Task

The ec2-describe-conversion-tasks command returns the status of an import task. Status values include the following:

- **active—**Your instance or volume is still importing.
- **cancelling—**Your instance or volume is still being canceled.
- **cancelled—**Your instance or volume is canceled.
- **completed—**Your instance or volume is ready to use.

  The imported instance is in the stopped state. You use `ec2-start-instance` to start it. For more information, see ec2-start-instances in the *Amazon EC2 Command Line Reference*.

To check the status of your import task

Use ec2-describe-conversion-tasks to return the status of the task. The syntax of the command is as follows:

```
ec2-describe-conversion-tasks task_id
```

Example

The following example enables you to see the status of your import instance task.

```
C:\> ec2-describe-conversion-tasks import-i-ffvko9js
```

Response 1

The following response shows that the `IMPORTINSTANCE` status is `active`, and 73747456 bytes out of 893968896 have been converted.

```
TaskType        IMPORTINSTANCE  TaskId  import-i-ffvko9js       ExpirationTime
  2011-06-07T13:30:50+00:00       Status  active  StatusMessage   Pending In
stanceID      i-17912579
DISKIMAGE       DiskImageFormat VMDK    DiskImageSize   893968896 VolumeSize
    12      AvailabilityZone        us-east-1       ApproximateBytesConverted
     73747456       Status  active StatusMessage    Pending
```

Response 2

The following response shows that the `IMPORTINSTANCE` status is `active`, at 7% progress, and the DISKIMAGE is completed.

```
TaskType        IMPORTINSTANCE  TaskId  import-i-ffvko9js       ExpirationTime
  2011-06-07T13:30:50+00:00       Status  active  StatusMessage   Progress: 7%
    InstanceID      i-17912579
DISKIMAGE       DiskImageFormat VMDK    DiskImageSize   893968896 VolumeId
    vol-9b59daf0    VolumeSize      12      AvailabilityZone        us-east-1
     ApproximateBytesConverted      893968896 Status  completed
```

Response 3

The following response shows that the `IMPORTINSTANCE` status is `completed`.

```
TaskType        IMPORTINSTANCE  TaskId  import-i-ffvko9js       ExpirationTime
  2011-06-07T13:30:50+00:00      Status  completed    InstanceID     i-17912579
DISKIMAGE       DiskImageFormat VMDK    DiskImageSize   893968896 VolumeId
    vol-9b59daf0    VolumeSize      12      AvailabilityZone        us-east-1
     ApproximateBytesConverted      893968896 Status  completed
```

> **Note**
> The `IMPORTINSTANCE` status is what you use to determine the final status. The `DISKIMAGE` status will be `completed` for a period of time before the `IMPORTINSTANCE` status is `completed`.

You can now use commands such as `ec2-stop-instance`, `ec2-start-instance`, `ec2-reboot-instance`, and `ec2-terminate-instance` to manage your instance. For more information, see the Amazon EC2 Command Line Reference

Importing Your Volumes into Amazon EBS

This section describes how to import your data storage into Amazon EBS, and then attach it to one of your existing EC2 instances. Amazon EC2 supports importing RAW, Virtual Hard Disk (VHD), and ESX Virtual Machine Disk (VMDK) disk formats.

> **Important**
> We recommend using Amazon EC2 security groups to limit network access to your imported instance. Configure a security group to allow only trusted EC2 instances and remote hosts to connect to RDP and other service ports. For more information about security groups, see Amazon EC2 Security Groups for Windows Instances (p. 398).

After you have exported your virtual machine from the virtualization environment, importing the volume to Amazon EBS is a single-step process. You create an import task and upload the volume.

To import a volume into Amazon EBS

1. Use ec2-import-volume to create a task that allows you to upload your volume into Amazon EBS. The syntax of the command is as follows:

   ```
   ec2-import-volume disk_image -f file_format -s volume_size -z availabil
   ity_zone -b s3_bucket_name -o owner -w secret_key
   ```

   The following example creates an import volume task for importing a volume to the us-east-1 region in the d availability zone.

   ```
   C:\> ec2-import-volume Win_2008_R1_EE_64.vmdk –f vmdk –s 25 -z us-east-1d
   -b myawsbucket -o AKIAIOSFODNN7EXAMPLE -w wJalrXUtnFEMI/K7MDENG/bPxRfi
   CYEXAMPLEKEY --region us-east-1 -o AKIAI44QH8DHBEXAMPLE -w je7MtGbCl
   wBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
   ```

The following is an example response.

```
Requesting volume size: 25 GB
Disk image format: Stream-optimized VMDK
Converted volume size: 26843545600 bytes (25.00 GiB)
Requested EBS volume size: 26843545600 bytes (25.00 GiB)
TaskType       IMPORTVOLUME    TaskId  import-vol-ffut5xv4     ExpirationTime
  2011-09-09T15:22:30+00:00       Status  active  StatusMessage   Pending
DISKIMAGE       DiskImageFormat VMDK    DiskImageSize   5070303744
VolumeSize      25      AvailabilityZone        us-east-1d      Approximate
BytesConverted       0
Creating new manifest at myawsbucket/0fd8fcf5-04d8-44ae-981f-
3c9f56d04520/Win_2008_R1_EE_64.vmdkmanifest.xml
Uploading the manifest file
Uploading 5070303744 bytes across 484 parts
0% |-------------------------------------------------| 100%
   |=================================================|
Done
```

Amazon EC2 returns a task ID that you use in the next step. In this example, the ID is `import-vol-ffut5xv4`.

2. Use ec2-describe-conversion-tasks to confirm that your volume imported successfully.

```
C:\> ec2-describe-conversion-tasks import-vol-ffut5xv4
TaskType       IMPORTVOLUME    TaskId  import-vol-ffut5xv4     ExpirationTime
  2011-09-09T15:22:30+00:00       Status  completed
DISKIMAGE       DiskImageFormat VMDK    DiskImageSize   5070303744
VolumeId        vol-365a385c    VolumeSize      25      AvailabilityZone
      us-east-1d      ApproximateBytesConverted       5070303744
```

The status in this example is `completed`, which means the import succeeded.

3. Use ec2-attach-volume to attach the Amazon EBS volume to one of your existing EC2 instances. The following example attaches the volume, `vol-2540994c`, to the `i-a149ec4a` instance on the device, `/dev/sde`.

```
C:\> ec2-attach-volume vol-2540994c -i i-a149ec4a -d xvde
ATTACHMENT vol-2540994c i-a149ec4a xvde attaching 2010-03-23T15:43:46+00:00
```

Resuming an Upload

Connectivity problems can interrupt an upload. When you resume an upload, Amazon EC2 automatically starts the upload from where it stopped. The following procedure steps you through determining how much of an upload succeeded and how to resume it.

To resume an upload

Use the task ID with ec2-resume-import to continue the upload. The command uses the HTTP `HEAD` action to determine where to resume.

```
ec2-resume-import disk_image -t task_id -o owner -w secret_key
```

Example

The following example resumes an import instance task.

```
C:\> ec2-resume-import Win_2008_R1_EE_64.vmdk -t import-i-ffni8aei -o AKIAIOS
FODNN7EXAMPLE -w wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

The following shows the output when the import instance task is complete:

```
Disk image size: 5070303744 bytes (4.72 GiB)
Disk image format: Stream-optimized VMDK
Converted volume size: 26843545600 bytes (25.00 GiB)
Requested EBS volume size: 26843545600 bytes (25.00 GiB)
Uploading 5070303744 bytes across 484 parts
0% |-------------------------------------------------| 100%
   |=================================================|
Done
Average speed was 10.316 MBps
The disk image for import-i-ffni8aei has been uploaded to Amazon S3
where it is being converted into an EC2 instance.  You may monitor the
progress of this task by running ec2-describe-conversion-tasks.  When
the task is completed, you may use ec2-delete-disk-image to remove the
image from S3.
```

Canceling an Upload

Use ec2-cancel-conversion-task to cancel an active import task. The task can be the upload of an instance or a volume. The command removes all artifacts of the import, including uploaded volumes or instances.

If the import is complete or still transferring the final disk image, the command fails and returns an exception similar to the following:

```
Client.CancelConversionTask Error: Failed to cancel conversion task import-i-
fh95npoc
```

To cancel an upload task

Use the task ID of the upload you want to delete with ec2-cancel-conversion-task.

Example

The following example cancels the upload associated with the task ID `import-i-fh95npoc`.

```
C:\> ec2-cancel-conversion-task import-i-fh95npoc
```

The output for a successful cancellation is similar to the following:

```
CONVERSION-TASK import-i-fh95npoc
```

You can use the ec2-describe-conversion-tasks command to check the status of the cancellation as in the following example:

```
C:\> ec2-describe-conversion-tasks import-i-fh95npoc
TaskType        IMPORTINSTANCE  TaskId  import-i-fh95npoc       ExpirationTime
  2010-12-20T18:36:39+00:00      Status  cancelled   InstanceID     i-825063ef
```

```
DISKIMAGE       DiskImageFormat VMDK    DiskImageSize   2671981568
VolumeSize      40      AvailabilityZone        us-east-1c  ApproximateBytesCon
verted       0       Status  cancelled
```

In this example, the status is `cancelled`. If the upload were still in process, the status would be `cancelling`.

Cleaning Up After an Upload

You can use ec2-delete-disk-image to remove the image file after it is uploaded. If you do not delete it, you will be charged for its storage in Amazon S3.

To delete a disk image

Use the task ID of the disk image you want to delete with `ec2-delete-disk-image`.

Example

The following example deletes the disk image associated with the task ID, `import-i-fh95npoc`.

```
C:\> ec2-delete-disk-image -t import-i-fh95npoc
```

The output for a successful cancellation is similar to the following:

```
DELETE-TASK import-i-fh95npoc
```

Step 5: Launch the instance in Amazon EC2

After you upload the VM to Amazon S3, the VM Import process automatically converts it into an Amazon EC2 instance and launches it as a stopped instance in the Amazon EC2 console. Before you can begin using the instance, you must start it. For more information about working with an Amazon EC2 instance, see Instance Lifecycle (p. 203).

To start the instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see Regions and Endpoints.
3. In the navigation pane, click **Instances**.
4. In the content pane, right-click the instance, select **Instance State**, and then click **Start**.

Exporting Amazon EC2 Instances

If you have previously imported an instance into Amazon EC2, you can use the command line tools to export that instance to Citrix Xen, Microsoft Hyper-V, or VMware vSphere. Exporting an instance that you previously imported is useful when you want to deploy a copy of your EC2 instance in your on-site virtualization environment.

If you're using VMware vSphere, you can also use the AWS Connector for vCenter to export a VM from Amazon EC2. For more information, see Exporting a Migrated Amazon EC2 Instance in the *AWS Management Portal for vCenter User Guide*.

Contents

- Export an Instance (p. 196)
- Cancel or Stop the Export of an Instance (p. 197)

Export an Instance

You can use the Amazon EC2 CLI to export an instance. If you haven't installed the CLI already, see Setting Up the Amazon EC2 Tools.

The ec2-create-instance-export-task command gathers all of the information necessary (e.g., instance ID; name of the Amazon S3 bucket that will hold the exported image; name of the exported image; VMDK, OVA, or VHD format) to properly export the instance to the selected virtualization format. The exported file is saved in the Amazon S3 bucket that you designate.

Note
When you export an instance, you are charged the standard Amazon S3 rates for the bucket where the exported VM is stored. In addition, a small charge reflecting temporary use of an Amazon EBS snapshot might appear on your bill. For more information about Amazon S3 pricing, see Amazon Simple Storage Service (S3) Pricing.

To export an instance

1. Create an Amazon S3 bucket for storing the exported instances. The Amazon S3 bucket must grant **Upload/Delete** and **View Permissions** access to the **vm-import-export@amazon.com** account. For more information, see Creating a Bucket and Editing Bucket Permissions in the *Amazon Simple Storage Service Console User Guide*.

   Note
   Instead of the **vm-import-export@amazon.com** account, you can use region-specific canonical IDs. The Amazon S3 bucket for the destination image must exist and must have WRITE and READ_ACP permissions granted to the following region-specific accounts using their canonical ID:

   - China (Beijing): 834bafd86b15b6ca71074df0fd1f93d234b9d5e848a2cb31f880c149003ce36f'
   - AWS GovCloud (US) : af913ca13efe7a94b88392711f6cfc8aa07c9d1454d4f190a624b126733a5602

     For more information, see Amazon Elastic Compute Cloud (Amazon EC2) in the *AWS GovCloud (US) User Guide*.
   - All other regions: c4d8eabf8db69dbe46bfe0e517100c554f01200b104d59cd408e777ba442a322

2. At a command prompt, type the following command:

```
ec2-create-instance-export-task instance_id –e target_environment –f
disk_image_format -c container_format –b s3_bucket
```

instance_id
The ID of the instance you want to export.

target_environment
VMware, Citrix, or Microsoft.

disk_image_format
VMDK for VMware or VHD for Microsoft Hyper-V and Citrix Xen.

container_format
Optionally set to OVA when exporting to VMware.

s3_bucket
The name of the Amazon S3 bucket to which you want to export the instance.

3. To monitor the export of your instance, at the command prompt, type the following command, where *task_id* is the ID of the export task:

```
ec2-describe-export-tasks task_id
```

Cancel or Stop the Export of an Instance

You can use the Amazon EC2 CLI to cancel or stop the export of an instance up to the point of completion. The ec2-cancel-export-task command removes all artifacts of the export, including any partially created Amazon S3 objects. If the export task is complete or is in the process of transferring the final disk image, the command fails and returns an error.

To cancel or stop the export of an instance

At the command prompt, type the following command, where *task_id* is the ID of the export task:

```
ec2-cancel-export-task task_id
```

Troubleshooting VM Import/Export

When importing or exporting a VM, most errors occur when you attempt to do something that isn't supported. To avoid these errors, read VM Import/Export Prerequisites (p. 169) before you begin an import or an export.

Errors

- AWS Error Code: InvalidParameter, AWS Error Message: Parameter disk-image-size=0 has an invalid format. (p. 198)
- Client.UnsupportedOperation: This instance has multiple volumes attached. Please remove additional volumes. (p. 198)
- Client.Unsupported: No bootable partition found. (Service: AmazonEC2; Status Code: 400; Error Code: Unsupported; Request ID: <RequestID>) (p. 198)
- ClientError: Footers not identical (p. 198)
- ClientError: Uncompressed data has invalid length. (p. 198)
- ERROR: Bucket <MyBucketName> is not in the <RegionName> region, it's in <RegionName>. (p. 198)
- ERROR: File uses unsupported compression algorithm 0. (p. 199)
- Error starting instances: Invalid value <instance ID> for instanceId. Instance does not have a volume attached at root (/dev/sda1). (p. 199)
- java.lang.OutOfMemoryError: Java heap space (p. 199)
- Service.InternalError: An internal error has occurred. Status Code: 500, AWS Service: AmazonEC2 (p. 199)
- A client error (MalformedPolicyDocument) occurred when calling the CreateRole operation: Syntax errors in policy. (p. 199)
- FirstBootFailure: This import request failed because the Windows instance failed to boot and establish network connectivity. (p. 200)
- Linux is not supported on the requested instance (p. 201)

AWS Error Code: InvalidParameter, AWS Error Message: Parameter disk-image-size=0 has an invalid format.

The image format you used is not supported.

Resolution

Retry using one of the supported image formats: RAW, VHD, or VMDK.

Client.UnsupportedOperation: This instance has multiple volumes attached. Please remove additional volumes.

The VM has multiple attached disks.

Resolution

Detach the extra drives and try again. If you need the data on the other volumes, copy the data to the root volume and try to export the VM again.

Client.Unsupported: No bootable partition found. (Service: AmazonEC2; Status Code: 400; Error Code: Unsupported; Request ID: *<RequestID>*)

The VM has a root volume that is GUID Partition Table (GPT) partitioned.

Resolution

GPT partitioned volumes are not supported by the VM Import/Export tools. Convert your VM's root volume to an MBR partition and then try importing the VM again.

ClientError: Footers not identical

You attempted to import a fixed or differencing VHD, or there was an error in creating the VHD.

Resolution

Export your VM again and retry importing it into Amazon EC2.

ClientError: Uncompressed data has invalid length.

The VMDK file is corrupted.

Resolution

You can try repairing or recreating the VMDK file, or use another one for your import.

ERROR: Bucket *<MyBucketName>* is not in the *<RegionName>* region, it's in *<RegionName>*.

The Amazon S3 bucket is not in the same region as the instance you want to import.

Resolution

Try adding the `--ignore-region-affinity` option, which ignores whether the bucket's region matches the region where the import task is created. You can also create an Amazon S3 bucket using the Amazon Simple Storage Service console and set the region to the region where you want to import the VM. Run the command again and specify the new bucket you just created.

ERROR: File uses unsupported compression algorithm 0.

The VMDK was created using OVA format instead of OVF format.

Resolution

Create the VMDK in OVF format.

Error starting instances: Invalid value *<instance ID>* for instanceId. Instance does not have a volume attached at root (/dev/sda1).

You attempted to start the instance before the VM import process and all conversion tasks were complete.

Resolution

Wait for the VM import process and all conversion tasks to completely finish, and then start the instance.

java.lang.OutOfMemoryError: Java heap space

There is not enough virtual memory available to launch Java, or the image you are trying to import is too large.

Resolution

If you allocate extra memory to Java, the extra memory will only apply to JVM, but if that setting is specified (explicitly for the EC2 command line tools) it will override the global settings. For example, you can use the following command to allocate 512 MB of extra memory to Java 'set EC2_JVM_ARGS=-Xmx512m'.

Service.InternalError: An internal error has occurred. Status Code: 500, AWS Service: AmazonEC2

You tried to import an instance that does not have a default VPC without specifying the subnet and Availability Zone.

Resolution

If you're importing an instance without a default VPC, be sure to specify the subnet and Availability Zone.

A client error (MalformedPolicyDocument) occurred when calling the CreateRole operation: Syntax errors in policy.

You forgot to include file:// before the policy document name.

Resolution

Include file:// before the policy document name (e.g., file://trust-policy.json).

FirstBootFailure: This import request failed because the Windows instance failed to boot and establish network connectivity.

When you import a VM using the `ec2-import-instance` command, the import task might stop before its completed, and then fail. To investigate what went wrong, you can use the ec2-describe-conversion-tasks command to describe the instance.

When you receive the FirstBootFailure error message, it means that your virtual disk image was unable to perform one of the following steps:

- Boot up and start Windows.
- Install Amazon EC2 networking and disk drivers.
- Use a DHCP-configured network interface to retrieve an IP address.
- Activate Windows using the Amazon EC2 Windows volume license.

The following best practices can help you to avoid Windows first boot failures:

- **Disable anti-virus and anti-spyware software and firewalls**. These types of software can prevent installing new Windows services or drivers or prevent unknown binaries from running. Software and firewalls can be re-enabled after importing.
- **Do not harden your operating system**. Security configurations, sometimes called hardening, can prevent unattended installation of Amazon EC2 drivers. There are numerous Windows configuration settings that can prevent import. These settings can be reapplied once imported.
- **Disable or delete multiple bootable partitions**. If your virtual machine boots and requires you to choose which boot partition to use, the import may fail.

This inability of the virtual disk image to boot up and establish network connectivity could be due to any of the following causes.

Causes

- TCP/IP networking and DHCP are not enabled (p. 200)
- A volume that Windows requires is missing from the virtual machine (p. 201)
- Windows always boots into System Recovery Options (p. 201)
- The virtual machine was created using a physical-to-virtual (P2V) conversion process (p. 201)
- Windows activation fails (p. 201)
- No bootable partition found (p. 201)

TCP/IP networking and DHCP are not enabled

Cause: For any Amazon EC2 instance, including those in Amazon VPC, TCP/IP networking and DHCP must be enabled. Within a VPC, you can define an IP address for the instance either before or after importing the instance. Do not set a static IP address before exporting the instance.

Resolution: Ensure that TCP/IP networking is enabled. For more information, see Setting up TCP/IP (Windows Server 2003) or Configuring TCP/IP (Windows Server 2008) at the Microsoft TechNet website.

Ensure that DHCP is enabled. For more information, see What is DHCP at the Microsoft TechNet web site.

A volume that Windows requires is missing from the virtual machine

Cause: Importing a VM into Amazon EC2 only imports the boot disk, all other disks must be detached and Windows must able to boot before importing the virtual machine. For example, Active Directory often stores the Active Directory database on the `D:\` drive. A domain controller cannot boot if the Active Directory database is missing or inaccessible.

Resolution: Detach any secondary and network disks attached to the Windows VM before exporting.

Move any Active Directory databases from secondary drives or partitions onto the primary Windows partition. For more information, see "Directory Services cannot start" error message when you start your Windows-based or SBS-based domain controller at the Microsoft Support website.

Windows always boots into System Recovery Options

Cause: Windows can boot into System Recovery Options for a variety of reasons, including when Windows is pulled into a virtualized environment from a physical machine, also known as P2V.

Resolution: Ensure that Windows boots to a login prompt before exporting and preparing for import.

Do not import virtualized Windows instances that have come from a physical machine.

The virtual machine was created using a physical-to-virtual (P2V) conversion process

Cause: A P2V conversion occurs when a disk image is created by performing the Windows installation process on a physical machine and then importing a copy of that Windows installation into a VM. VMs that are created as the result of a P2V conversion are not supported by Amazon EC2 VM import. Amazon EC2 VM import only supports Windows images that were natively installed inside the source VM.

Resolution: Install Windows in a virtualized environment and migrate your installed software to that new VM.

Windows activation fails

Cause: During boot, Windows will detect a change of hardware and attempt activation. During the import process we attempt to switch the licensing mechanism in Windows to a volume license provided by Amazon Web Services. However, if the Windows activation process does not succeed, then the import will not succeed.

Resolution: Ensure that the version of Windows you are importing supports volume licensing. Beta or preview versions of Windows might not.

No bootable partition found

Cause: During the import process of a virtual machine, we could not find the boot partition.

Resolution: Ensure that the disk you are importing has the boot partition. We do not support multi-disk import.

Linux is not supported on the requested instance

Cause: Linux import is only supported on specific instance types. You attempted to import an unsupported instance type.

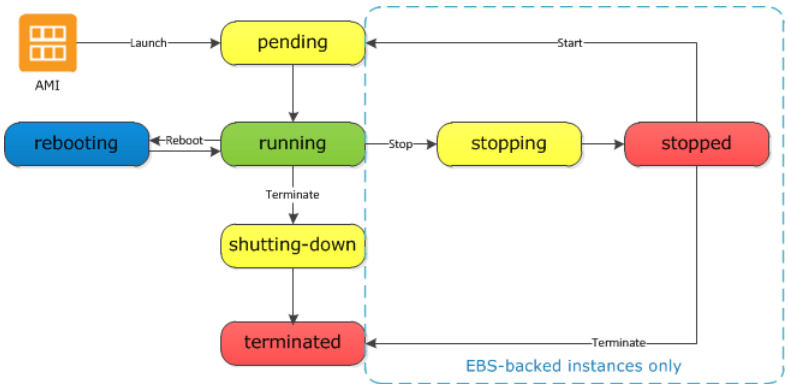
Resolution: Retry using one of the supported instance types.

- General purpose: `t2.micro` | `t2.small` | `t2.medium` | `m3.medium` | `m3.large` | `m3.xlarge` | `m3.2xlarge`
- Compute optimized: `c3.large` | `c3.xlarge` | `c3.2xlarge` | `c3.4xlarge` | `cc2.8xlarge`
- Memory optimized: `cr1.8xlarge`
- Storage optimized: `hi1.4xlarge` | `hs1.8xlarge` | `i2.xlarge` | `i2.2xlarge` | `i2.4xlarge`
- GPU: `cg1.4xlarge`

Instance Lifecycle

By working with Amazon EC2 to manage your instances from the moment you launch them through their termination, you ensure that your customers have the best possible experience with the applications or sites that you host on your instances.

The following illustration represents the transitions between instance states. Notice that you can't stop and start an instance store-backed instance. For more information about instance store-backed instances, see Storage for the Root Device (p. 53).



Instance Launch

When you launch an instance, it enters the `pending` state. The instance type that you specified at launch determines the hardware of the host computer for your instance. We use the Amazon Machine Image (AMI) you specified at launch to boot the instance. After the instance is ready for you, it enters the `running` state. You can connect to your running instance and use it the way that you'd use a computer sitting in front of you.

As soon as your instance transitions to the `running` state, you're billed for each hour or partial hour that you keep the instance running; even if the instance remains idle and you don't connect to it.

For more information, see Launch Your Instance (p. 206) and Connecting to Your Windows Instance Using RDP (p. 216).

Instance Stop and Start (Amazon EBS-backed instances only)

If your instance fails a status check or is not running your applications as expected, and if the root volume of your instance is an Amazon EBS volume, you can stop and start your instance to try to fix the problem.

When you stop your instance, it enters the `stopping` state, and then the `stopped` state. We don't charge hourly usage or data transfer fees for your instance after you stop it, but we do charge for the storage for any Amazon EBS volumes. While your instance is in the `stopped` state, you can modify certain attributes of the instance, including the instance type.

When you start your instance, it enters the `pending` state, and we move the instance to a new host computer. Therefore, when you stop and start your instance, you'll lose any data on the instance store volumes on the previous host computer.

If your instance is running in EC2-Classic, it receives a new private IP address, which means that an Elastic IP address (EIP) associated with the private IP address is no longer associated with your instance. If your instance is running in EC2-VPC, it retains its private IP address, which means that an EIP associated with the private IP address or network interface is still associated with your instance.

Each time you transition an instance from `stopped` to `running`, we charge a full instance hour, even if these transitions happen multiple times within a single hour.

For more information, see Stop and Start Your Instance (p. 218).

Instance Reboot

You can reboot your instance using the Amazon EC2 console, the Amazon EC2 CLI, and the Amazon EC2 API. We recommend that you use Amazon EC2 to reboot your instance instead of running the operating system reboot command from your instance.

Rebooting an instance is equivalent to rebooting an operating system; the instance remains on the same host computer and maintains its public DNS name, private IP address, and any data on its instance store volumes. It typically takes a few minutes for the reboot to complete, but the time it takes to reboot depends on the instance configuration.

Rebooting an instance doesn't start a new instance billing hour.

For more information, see Reboot Your Instance (p. 222).

Instance Retirement

An instance is scheduled to be retired when AWS detects irreparable failure of the underlying hardware hosting the instance. When an instance reaches its scheduled retirement date, it is stopped or terminated by AWS. If your instance root device is an Amazon EBS volume, the instance is stopped, and you can start it again at any time. If your instance root device is an instance store volume, the instance is terminated, and cannot be used again.

For more information, see Instance Retirement (p. 222).

Instance Termination

When you've decided that you no longer need an instance, you can terminate it. As soon as the status of an instance changes to `shutting-down` or `terminated`, you stop incurring charges for that instance.

Note that if you enable termination protection, you can't terminate the instance using the console, CLI, or API.

After you terminate an instance, it remains visible in the console for a short while, and then the entry is deleted. You can also describe a terminated instance using the CLI and API. You can't connect to or recover a terminated instance.

Each Amazon EBS-backed instance supports the `InstanceInitiatedShutdownBehavior` attribute, which controls whether the instance stops or terminates when you initiate a shutdown from within the instance itself. The default behavior is to stop the instance. You can modify the setting of this attribute while the instance is running or stopped.

Each Amazon EBS volume supports the `DeleteOnTermination` attribute, which controls whether the volume is deleted or preserved when you terminate the instance it is attached to. The default is to delete the root device volume and preserve any other EBS volumes.

For more information, see Terminate Your Instance (p. 224).

Differences Between Reboot, Stop, and Terminate

The following table summarizes the key differences between rebooting, stopping, and terminating your instance.

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Terminate
Host computer	The instance stays on the same host computer	The instance runs on a new host computer	None
Private and public IP addresses	These addresses stay the same	EC2-Classic: The instance gets new private and public IP addresses EC2-VPC: The instance keeps its private IP address. The instance gets a new public IP address, unless it has an Elastic IP address (EIP), which doesn't change during a stop/start.	None
Elastic IP addresses (EIP)	The EIP remains associated with the instance	EC2-Classic: The EIP is disassociated from the instance EC2-VPC: The EIP remains associated with the instance	The EIP is disassociated from the instance

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Terminate
Instance store volumes	The data is preserved	The data is erased	The data is erased
Root device volume	The volume is preserved	The volume is preserved	The volume is deleted by default
Billing	The instance billing hour doesn't change.	You stop incurring charges for an instance as soon as its state changes to `stopping`. Each time an instance transitions from `stopped` to `pending`, we start a new instance billing hour.	You stop incurring charges for an instance as soon as its state changes to `shutting-down`.

Note that operating system shutdown commands always terminate an instance store-backed instance. You can control whether operating system shutdown commands stop or terminate an Amazon EBS-backed instance. For more information, see Changing the Instance Initiated Shutdown Behavior (p. 227).

Launch Your Instance

An instance is a virtual server in the AWS cloud. You launch an instance from an Amazon Machine Image (AMI). The AMI provides the operating system, application server, and applications for your instance.

When you sign up for AWS, you can get started with Amazon EC2 for free using the AWS Free Tier. You can either leverage the free tier to launch and use a micro instance for free for 12 months. If you launch an instance that is not within the free tier, you incur the standard Amazon EC2 usage fees for the instance. For more information, see the Amazon EC2 Pricing.

You can launch an instance using the following methods.

Method	Documentation
Use the Amazon EC2 console with an AMI that you select	Launching an Instance (p. 207)
Use the Amazon EC2 console to launch an instance using an existing instance as a template	Launching an Instance Using an Existing Instance as a Template (p. 212)
Use the Amazon EC2 console with an AMI that you purchased from the AWS Marketplace	Launching an AWS Marketplace Instance (p. 213)
Use the AWS CLI with an AMI that you select	Using Amazon EC2 through the AWS CLI
Use the Amazon EC2 CLI with an AMI that you select	Launching an Instance Using the Amazon EC2 CLI
Use the AWS Tools for Windows PowerShell with an AMI that you select	Amazon EC2 from the AWS Tools for Windows PowerShell

After you launch your instance, you can connect to it and use it. To begin, the instance state is `pending`. When the instance state is `running`, the instance has started booting. There might be a short time before you can connect to the instance. The instance receives a public DNS name that you can use to contact the instance from the Internet. The instance also receives a private DNS name that other instances within the same Amazon EC2 network (EC2-Classic or EC2-VPC) can use to contact the instance. For more information about connecting to your instance, see Connecting to Your Windows Instance Using RDP (p. 216).

When you are finished with an instance, be sure to terminate it. For more information, see Terminate Your Instance (p. 224).

Launching an Instance

Before you launch your instance, be sure that you are set up. For more information, see Setting Up with Amazon EC2 (p. 14).

Your AWS account might support both the EC2-Classic and EC2-VPC platforms, depending on when you created your account and which regions you've used. To find out which platform your account supports, see Supported Platforms (p. 455). If your account supports EC2-Classic, you can launch an instance into either platform. If your account supports EC2-VPC only, you can launch an instance into a VPC only.

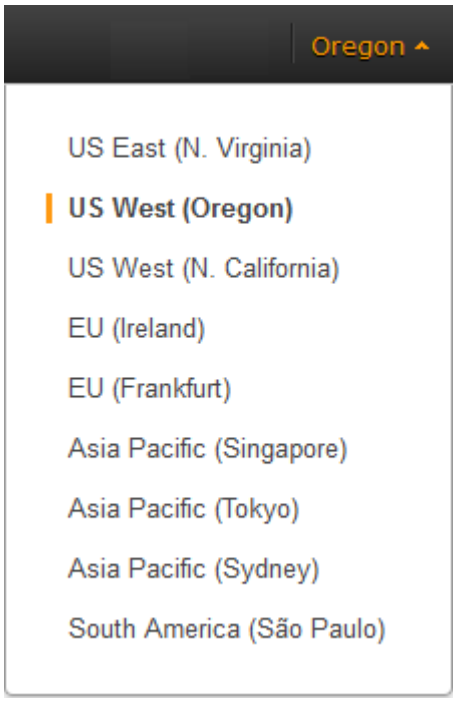
Important
When you launch an instance that's not within the AWS Free Tier, you are charged for the time that the instance is running, even if it remains idle.

Launching Your Instance from an AMI

When you launch an instance, you must select a configuration, known as an Amazon Machine Image (AMI). An AMI contains the information required to create a new instance. For example, an AMI might contain the software required to act as a web server: for example, Windows, Apache, and your web site.

To launch an instance

1. Open the Amazon EC2 console.
2. In the navigation bar at the top of the screen, the current region is displayed. Select the region for the instance. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. Select the region that meets your needs. For more information, see Resource Locations (p. 605).



3. From the Amazon EC2 console dashboard, click **Launch Instance**.
4. On the **Choose an Amazon Machine Image (AMI)** page, choose an AMI as follows:

   a. Select the type of AMI to use in the left pane:

      Quick Start
      A selection of popular AMIs to help you get started quickly. To ensure that you select an AMI that is eligible for the free tier, click **Free tier only** in the left pane. (Notice that these AMIs are marked **Free tier eligible**.)

      My AMIs
      The private AMIs that you own, or private AMIs that have been shared with you.

      AWS Marketplace
      An online store where you can buy software that runs on AWS, including AMIs. For more information about launching an instance from the AWS Marketplace, see Launching an AWS Marketplace Instance (p. 213).

      Community AMIs
      The AMIs that AWS community member have made available for others to use. To filter the list of AMIs by operating system, select the appropriate check box under **Operating system**. You can also filter by architecture and root device type.

   b. Check the **Root device type** listed for each AMI. Notice which AMIs are the type that you need, either `ebs` (backed by Amazon EBS) or `instance-store` (backed by instance store). For more information, see Storage for the Root Device (p. 53).
   c. Check the **Virtualization type** listed for each AMI. Notice which AMIs are the type that you need, either `hvm` or `paravirtual`. For example, some instance types require HVM.
   d. Choose an AMI that meets your needs, and then click **Select**.

5. On the **Choose an Instance Type** page, select the hardware configuration and size of the instance to launch. Larger instance types have more CPU and memory. For more information, see Instance Types (p. 96).

To remain eligible for the free tier, select the **t2.micro** instance type. For more information, see T2 Instances (p. 99).

By default, the wizard displays current generation instance types, and selects the first available instance type based on the AMI that you selected. To view previous generation instance types, select **All generations** from the filter list.

> **Tip**
> If you are new to AWS and would like to set up an instance quickly for testing purposes, you can click **Review and Launch** at this point to accept default configuration settings, and launch your instance. Otherwise, to configure your instance further, click **Next: Configure Instance Details**.

6. On the **Configure Instance Details** page, change the following settings as necessary (expand **Advanced Details** to see all the settings), and then click **Next: Add Storage**:

   - **Number of instances**: Enter the number of instances to launch.
   - **Purchasing option**: Select **Request Spot Instances** to launch a Spot instance. For more information, see Spot Instances (p. 122).
   - Your account may support the EC2-Classic and EC2-VPC platforms, or EC2-VPC only. To find out which platform your account supports, see Supported Platforms (p. 455). If your account supports EC2-VPC only, you can launch your instance into your default VPC or a nondefault VPC. Otherwise, you can launch your instance into EC2-Classic or a nondefault VPC.

     > **Note**
     > You must launch a T2 instance into a VPC. If you don't have a VPC, you can let the wizard create one for you.

     To launch into EC2-Classic:

     - **Network**: Select **Launch into EC2-Classic**.
     - **Availability Zone**: Select the Availability Zone to use. To let AWS choose an Availability Zone for you, select **No preference**.

     To launch into a VPC:

     - **Network**: Select the VPC, or to create a new VPC, click **Create new VPC** to go the Amazon VPC console. When you have finished, return to the wizard and click **Refresh** to load your VPC in the list.
     - **Subnet**: Select the subnet into which to launch your instance. If your account is EC2-VPC only, select **No preference** to let AWS choose a default subnet in any Availability Zone. To create a new subnet, click **Create new subnet** to go to the Amazon VPC console. When you are done, return to the wizard and click **Refresh** to load your subnet in the list.
     - **Auto-assign Public IP**: Specify whether your instance receives a public IP address. By default, instances in a default subnet receive a public IP address and instances in a nondefault subnet do not. You can select **Enable** or **Disable** to override the subnet's default setting. For more information, see Public IP Addresses and External DNS Hostnames (p. 475).
   - **Domain join directory**: Select the AWS Directory Service directory (domain) to which your Windows instance is joined. The directory must be in the same VPC that you selected for your instance. If you select a domain, you must select an IAM role. For more information, see Joining a Windows Instance to an AWS Directory Service Domain (p. 291).
   - **IAM role**: If applicable, select an AWS Identity and Access Management (IAM) role to associate with the instance. For more information, see IAM Roles for Amazon EC2 (p. 442).
   - **Shutdown behavior**: Select whether the instance should stop or terminate when shut down. For more information, see Changing the Instance Initiated Shutdown Behavior (p. 227).
   - **Enable termination protection**: Select this check box to prevent accidental termination. For more information, see Enabling Termination Protection for an Instance (p. 226).

- **Monitoring**: Select this check box to enable detailed monitoring of your instance using Amazon CloudWatch. Additional charges apply. For more information, see Monitoring Your Instances with CloudWatch (p. 330).
- **EBS-Optimized instance**: An Amazon EBS-optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. If the instance type supports this feature, select this check box to enable it. Additional charges apply. For more information, see Amazon EBS–Optimized Instances (p. 555).
- **Tenancy**: If you are launching your instance into a VPC, you can select **Dedicated tenancy** to run your instance on isolated, dedicated hardware. Additional charges apply. For more information, see Dedicated Instances in the *Amazon VPC User Guide*.
- **Network interfaces**: If you are launching an instance into a VPC and you did not select **No Preference** for your subnet, you can specify up to two network interfaces in the wizard. Click **Add IP** to assign more than one IP address to the selected interface. For more information about network interfaces, see Elastic Network Interfaces (ENI) (p. 492). If you selected the **Public IP** check box above, you can only assign a public IP address to a single, new network interface with the device index of eth0. For more information, see Assigning a Public IP Address (p. 479).
- **Kernel ID**: (Only valid for paravirtual (PV) AMIs) Select **Use default** unless you want to use a specific kernel.
- **RAM disk ID**: (Only valid for paravirtual (PV) AMIs) Select **Use default** unless you want to use a specific RAM disk. If you have selected a kernel, you may need to select a specific RAM disk with the drivers to support it.
- **Placement group**: A placement group is a logical grouping for your cluster instances. Select an existing placement group, or create a new one. This option is only available if you've selected an instance type that supports placement groups. For more information, see Placement Groups (p. 504).
- **User data**: You can specify user data to configure an instance during launch, or to run a configuration script. To attach a file, select the **As file** option and browse for the file to attach.

7. On the **Add Storage** page, you can specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume). You can change the following options, then click **Next: Tag Instance** when you have finished:

   - **Type**: Select instance store or Amazon EBS volumes to associate with your instance. The type of volume available in the list depends on the instance type you've chosen. For more information, see Amazon EC2 Instance Store (p. 577) and Amazon EBS Volumes (p. 518).
   - **Device**: Select from the list of available device names for the volume.
   - **Snapshot**: Enter the name or ID of the snapshot from which to restore a volume. You can also search for public snapshots by typing text into the **Snapshot** field. Snapshot descriptions are case-sensitive.
   - **Size**: For Amazon EBS-backed volumes, you can specify a storage size. Note that even if you have selected an AMI and instance that are eligible for the free tier, you need to keep under 30 GiB of total storage to stay within the free tier.

     Note

     The following Amazon EBS volume considerations apply to Windows boot volumes:

     - Windows 2003 instances will not boot if the boot volume is 2 TiB (2048 GiB) or greater
     - Windows boot volumes must use an MBR partition table, which limits the usable space to 2 TiB, regardless of volume size
     - Windows boot volumes 2 TiB (2048 GiB) or greater that have been converted to use a dynamic MBR partition table display an error when examined with Disk Manager

     The following Amazon EBS volume considerations apply to Windows data (non-boot) volumes:

     - Windows volumes 2 TiB (2048 GiB) or greater must use a GPT partition table to access the entire volume

Note
If you increase the size of your root volume at this point (or any other volume created from a snapshot), you need to extend the file system on that volume in order to use the extra space. For more information about extending your file system after your instance has launched, see Expanding the Storage Space of an EBS Volume on Windows (p. 544).

- **Volume Type**: For Amazon EBS volumes, select either a General Purpose (SSD), Provisioned IOPS (SSD), or Magnetic volume. For more information, see Amazon EBS Volume Types (p. 520).

  Note
  If you select a Magnetic boot volume, you'll be prompted when you complete the wizard to make General Purpose (SSD) volumes the default boot volume for this instance and future console launches. (This preference persists in the browser session, and does not affect AMIs with Provisioned IOPS (SSD) boot volumes.) We recommended that you make General Purpose (SSD) volumes the default because they provide a much faster boot experience and they are the optimal volume type for most workloads. For more information, see Amazon EBS Volume Types (p. 520).

  Note
  Some AWS accounts created before 2012 might have access to Availability Zones in us-east-1, us-west-1, or ap-northeast-1 that do not support Provisioned IOPS (SSD) volumes. If you are unable to create a Provisioned IOPS (SSD) volume (or launch an instance with a Provisioned IOPS (SSD) volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports Provisioned IOPS (SSD) volumes by creating a 4 GiB Provisioned IOPS (SSD) volume in that zone.

- **IOPS**: If you have selected a Provisioned IOPS (SSD) volume type, then you can enter the number of I/O operations per second (IOPS) that the volume can support.
- **Delete on Termination**: For Amazon EBS volumes, select this check box to delete the volume when the instance is terminated. For more information, see Preserving Amazon EBS Volumes on Instance Termination (p. 227).
- **Encrypted**: Select this check box to encrypt new Amazon EBS volumes. Amazon EBS volumes that are restored from encrypted snapshots are automatically encrypted. Encrypted volumes may only be attached to supported instance types (p. 559).

  Note
  Encrypted boot volumes are not supported at this time.

8. On the **Tag Instance** page, specify tags (p. 609) for the instance by providing key and value combinations. Click **Create Tag** to add more than one tag to your resource. Click **Next: Configure Security Group** when you are done.
9. On the **Configure Security Group** page, use a security group to define firewall rules for your instance. These rules specify which incoming network traffic is delivered to your instance. All other traffic is ignored. (For more information about security groups, see Amazon EC2 Security Groups for Windows Instances (p. 398).) Select or create a security group as follows, and then click **Review and Launch**.

   To select an existing security group:

   1. Click **Select an existing security group**. Your security groups are displayed. (If you are launching into EC2-Classic, these are security groups for EC2-Classic. If you are launching into a VPC, these are security group for that VPC.)
   2. Select a security group from the list.
   3. (Optional) You can't edit the rules of an existing security group, but you can copy them to a new group by clicking **Copy to new**. Then you can add rules as described in the next procedure.

   To create a new security group:

1. Click **Create a new security group**. The wizard automatically defines the launch-wizard-*x* security group.
2. (Optional) You can edit the name and description of the security group.
3. The wizard automatically defines an inbound rule to allow to you connect to your instance over SSH (port 22) for Linux or RDP (port 3389) for Windows.

   Caution
   This rule enables all IP addresses (`0.0.0.0/0`) to access your instance over the specified port. This is acceptable for this short exercise, but it's unsafe for production environments. You should authorize only a specific IP address or range of addresses to access your instance.
4. You can add rules to suit your needs. For example, if your instance is a web server, open ports 80 (HTTP) and 443 (HTTPS) to allow Internet traffic.

   To add a rule, click **Add Rule**, select the protocol to open to network traffic, and then specify the source. Select **My IP** from the **Source** list to let the wizard add your computer's public IP address. However, if you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

10. On the **Review Instance Launch** page, check the details of your instance, and make any necessary changes by clicking the appropriate **Edit** link.

    When you are ready, click **Launch**.
11. In the **Select an existing key pair or create a new key pair** dialog box, you can choose an existing key pair, or create a new one. For example, select **Choose an existing key pair**, then select the key pair you created when getting set up.

    To launch your instance, select the acknowledgment check box, then click **Launch Instances**.

    Important
    If you select the **Proceed without key pair** option, you won't be able to connect to the instance unless you choose an AMI that is configured to allow users another way to log in.
12. (Optional) You can create a status check alarm for the instance (additional fees may apply). (If you're not sure, you can always add one later.) On the confirmation screen, click **Create status check alarms** and follow the directions. For more information, see Creating and Editing Status Check Alarms (p. 325).
13. If the instance state immediately goes to `terminated` instead of `running`, you can get information about why the instance didn't launch. For more information, see Instance terminates immediately (p. 691).

Launching an Instance Using an Existing Instance as a Template

The Amazon EC2 console provides a **Launch More Like This** wizard option that enables you to use a current instance as a template for launching other instances. This option automatically populates the Amazon EC2 launch wizard with certain configuration details from the selected instance.

Note
The **Launch More Like This** wizard option does not clone your selected instance; it only replicates some configuration details. To create a copy of your instance, first create an AMI from it, then launch more instances from the AMI.

The following configuration details are copied from the selected instance into the launch wizard:

- AMI ID

- Instance type
- Availability Zone, or the VPC and subnet in which the selected instance is located
- Public IP address. If the selected instance currently has a public IP address, the new instance receives a public IP address - regardless of the selected instance's default public IP address setting. For more information about public IP addresses, see Public IP Addresses and External DNS Hostnames (p. 475).
- Placement group, if applicable
- IAM role associated with the instance, if applicable
- Shutdown behavior setting (stop or terminate)
- Termination protection setting (true or false)
- CloudWatch monitoring (enabled or disabled)
- Amazon EBS-optimization setting (true or false)
- Tenancy setting, if launching into a VPC (shared or dedicated)
- Kernel ID and RAM disk ID, if applicable
- User data, if specified
- Tags associated with the instance, if applicable
- Security groups associated with the instance
- Association information. If the selected instance is associated with a configuration file, the same file is automatically associated with new instance. If the configuration file includes a domain join configuration, the new instance will be joined to the same domain. For more information about joining a domain, see Joining a Windows Instance to an AWS Directory Service Domain (p. 291).

The following configuration details are not copied from your selected instance; instead, the wizard applies their default settings or behavior:

- (VPC only) Number of network interfaces: The default is one network interface, which is the primary network interface (eth0).
- Storage: The default storage configuration is determined by the AMI and the instance type.

To use your current instance as a template

1. On the Instances page, select the instance you want to use.
2. Click **Actions**, and select **Launch More Like This**.
3. The launch wizard opens on the **Review Instance Launch** page. You can check the details of your instance, and make any necessary changes by clicking the appropriate **Edit** link.

   When you are ready, click **Launch** to select a key pair and launch your instance.

Launching an AWS Marketplace Instance

You can subscribe to an AWS Marketplace product and launch an instance from the product's AMI using the Amazon EC2 launch wizard. For more information about paid AMIs, see Paid AMIs (p. 64). To cancel your subscription after launch, you first have to terminate all instances running from it. For more information, see Managing Your AWS Marketplace Subscriptions (p. 68).

To launch an instance from the AWS Marketplace using the launch wizard

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the Amazon EC2 dashboard, click **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, select the **AWS Marketplace** category on the left. Find a suitable AMI by browsing the categories, or using the search functionality. Click **Select** to choose your product.

4. A dialog displays an overview of the product you've selected. You can view the pricing information, as well as any other information that the vendor has provided. When you're ready, click **Continue**.

   Note
   You are not charged for using the product until you have launched an instance with the AMI. Take note of the pricing for each supported instance type, as you will be prompted to select an instance type on the next page of the wizard.

5. On the **Choose an Instance Type** page, select the hardware configuration and size of the instance to launch. When you're done, click **Next: Configure Instance Details**.
6. On the next pages of the wizard, you can configure your instance, add storage, and add tags. For more information about the different options you can configure, see Launching an Instance (p. 207). Click **Next** until you reach the **Configure Security Group** page.

   The wizard creates a new security group according to the vendor's specifications for the product. The security group may include rules that allow all IP addresses (`0.0.0.0/0`) access on SSH (port 22) on Linux or RDP (port 3389) on Windows. We recommend that you adjust these rules to allow only a specific address or range of addresses to access your instance over those ports.

   When you are ready, click **Review and Launch**.
7. On the **Review Instance Launch** page, check the details of the AMI from which you're about to launch the instance, as well as the other configuration details you set up in the wizard. When you're ready, click **Launch** to choose or create a key pair, and launch your instance.
8. Depending on the product you've subscribed to, the instance may take a few minutes or more to launch. You are first subscribed to the product before your instance can launch. If there are any problems with your credit card details, you will be asked to update your account details. When the launch confirmation page displays, click **View Instances** to go to the Instances page.

   Note
   You are charged the subscription price as long as your instance is running, even if it is idle. If your instance is stopped, you may still be charged for storage.

9. When your instance is in the **running** state, you can connect to it. To do this, select your instance in the list and click **Connect**. Follow the instructions in the dialog. For more information about connecting to your instance, see Connecting to Your Windows Instance Using RDP (p. 216).

   Important
   Check the vendor's usage instructions carefully, as you may need to use a specific user name to log in to the instance. For more information about accessing your subscription details, see Managing Your AWS Marketplace Subscriptions (p. 68).

Launching an AWS Marketplace AMI Instance Using the API and CLI

To launch instances from AWS Marketplace products using the API or command line tools, first ensure that you are subscribed to the product. You can then launch an instance with the product's AMI ID using the following methods:

Method	Documentation
AWS CLI	Use the run-instances command, or see the following topic for more information: Launching an Instance.
Amazon EC2 CLI	Use the ec2-run-instances command, or see the following topic for more information: Launching an Instance Using the Amazon EC2 CLI.

Method	Documentation
AWS Tools for Windows PowerShell	Use the New-EC2Instance command, or see the following topic for more information: Launch an Amazon EC2 Instance Using Windows PowerShell
Query API	Use the RunInstances request.

Connecting to Your Windows Instance Using RDP

After you launch your instance, you can connect to it and use it the way that you'd use a computer sitting in front of you.

If you receive an error while attempting to connect to your instance, see Troubleshooting Windows Instances (p. 687).

If you need to connect to a Linux instance, see Connect to Your Linux Instance in the *Amazon EC2 User Guide for Linux Instances*.

Prerequisites

- **Install an RDP client**
  Your Windows computer includes an RDP client by default. You can check for an RDP client by typing **mstsc** at a Command Prompt window. If your computer doesn't recognize this command, see the Microsoft Windows home page and search for the download for Remote Desktop Connection. For Mac OS X, you can use the Microsoft Remote Desktop app from the Apple App Store, or the Microsoft's Remote Desktop Connection Client from the Microsoft website. For Linux, you can use rdesktop.

  Important
  Mac OS X users: If you are connecting to a Windows 2012 R2 instance, the Remote Desktop Connection client from the Microsoft website may not work. Use the Microsoft Remote Desktop app from the Apple App Store instead.
- **Get the ID of the instance**
  You can get the ID of your instance using the Amazon EC2 console (from the **Instance ID** column). If you prefer, you can use the describe-instances (AWS CLI) or ec2-describe-instances (Amazon EC2 CLI) command.
- **Get the public DNS name of the instance**
  You can get the public DNS for your instance using the Amazon EC2 console (check the **Public DNS** column; if this column is hidden, click the **Show/Hide** icon and select **Public DNS**). If you prefer, you can use the describe-instances (AWS CLI) or ec2-describe-instances (Amazon EC2 CLI) command.
- **Locate the private key**
  You'll need the fully-qualified path of the `.pem` file for the key pair that you specified when you launched the instance.
- **Enable inbound RDP traffic from your IP address to your instance**
  Ensure that the security group associated with your instance allows incoming RDP traffic from your IP address. For more information, see Authorizing Inbound Traffic for Your Windows Instances (p. 448).

  Important
  Your default security group does not allow incoming RDP traffic by default.
- For the best experience using Internet Explorer, run the latest version.

Connect to Your Windows Instance

To connect to a Windows instance, you must retrieve the initial administrator password and then specify this password when you connect to your instance using Remote Desktop.

Note
If you've joined your instance to a domain, you can connect to your instance using domain credentials you've defined in AWS Directory Service. For more information about connecting to an instance in a domain, see Connecting To Your Instance Using Domain Credentials (p. 297).

The name of the administrator account depends on the language of the operating system. For example, for English, it's Administrator, for French it's Administrateur, and for Portuguese it's Administrador. For more information, see Localized Names for Administrator Account in Windows in the Microsoft TechNet Wiki.

Windows instances are limited to two simultaneous remote connections at one time. If you attempt a third connection, an error will occur. For more information, see Configure the Number of Simultaneous Remote Connections Allowed for a Connection.

To connect to your Windows instance using an RDP client

1. In the Amazon EC2 console, select the instance, and then choose **Connect**.
2. In the **Connect To Your Instance** dialog box, choose **Get Password** (it will take a few minutes after the instance is launched before the password is available).
3. Choose **Browse** and navigate to the private key file you created when you launched the instance. Select the file and choose **Open** to copy the entire contents of the file into contents box.
4. Choose **Decrypt Password**. The console displays the default administrator password for the instance in the **Connect To Your Instance** dialog box, replacing the link to **Get Password** shown previously with the actual password.
5. Record the default administrator password, or copy it to the clipboard. You need this password to connect to the instance.
6. Choose **Download Remote Desktop File**. Your browser prompts you to either open or save the .rdp file. Either option is fine. When you have finished, you can choose **Close** to dismiss the **Connect To Your Instance** dialog box.

   - If you opened the .rdp file, you'll see the **Remote Desktop Connection** dialog box.
   - If you saved the .rdp file, navigate to your downloads directory, and open the .rdp file to display the dialog box.

7. You may get a warning that the publisher of the remote connection is unknown. If you are using **Remote Desktop Connection** from a Windows PC, choose **Connect** to connect to your instance. If you are using **Microsoft Remote Desktop** on a Mac, skip the next step.
8. When prompted, log in to the instance, using the administrator account for the operating system and the password that you recorded or copied previously. If your **Remote Desktop Connection** already has an administrator account set up, you might have to choose the **Use another account** option and enter the user name and password manually.

   Note
   Sometimes copying and pasting content can corrupt data. If you encounter a "Password Failed" error when you log in, try typing in the password manually.

9. Due to the nature of self-signed certificates, you may get a warning that the security certificate could not be authenticated. Use the following steps to verify the identity of the remote computer, or simply choose **Yes** or **Continue** to continue if you trust the certificate.

   a. If you are using **Remote Desktop Connection** from a Windows PC, choose **View certificate**. If you are using **Microsoft Remote Desktop** on a Mac, choose **Show Certificate**.
   b. Choose the **Details** tab, and scroll down to the **Thumbprint** entry on a Windows PC, or the **SHA1 Fingerprints** entry on a Mac. This is the unique identifier for the remote computer's security certificate.

c. In the Amazon EC2 console, select the instance, choose **Actions**, and then choose **Get System Log**.
d. In the system log output, look for an entry labeled `RDPCERTIFICATE-THUMBPRINT`. If this value matches the thumbprint or fingerprint of the certificate, you have verified the identity of the remote computer.
e. If you are using **Remote Desktop Connection** from a Windows PC, return to the **Certificate** dialog box and choose **OK**. If you are using **Microsoft Remote Desktop** on a Mac, return to the **Verify Certificate** and choose **Continue**.
f. If you are using **Remote Desktop Connection** from a Windows PC, choose **Yes** in the **Remote Desktop Connection** window to connect to your instance. If you are using **Microsoft Remote Desktop** on a Mac, log in to the instance as prompted, using the default **Administrator** account and the default administrator password that you recorded or copied previously.

Note
On a Mac, you may need to switch spaces to see the **Microsoft Remote Desktop** login screen. For more information on spaces, see http://support.apple.com/kb/PH14155.

After you connect, we recommend that you do the following:

- Change the administrator password from the default value. You change the password while logged on to the instance itself, just as you would on any other Windows Server.
- Create another user account with administrator privileges on the instance. Another account with administrator privileges is a safeguard if you forget the administrator password or have a problem with the administrator account.

Transfer Files to Windows Server Instances

You can work with your Windows instance the same way that you would work with any Windows server. For example, you can transfer files between a Windows instance and your local computer using the local file sharing feature of the Microsoft Remote Desktop Connection software. If you enable this option, you can access your local files from your Windows instances. You can access local files on hard disk drives, DVD drives, portable media drives, and mapped network drives. For more information about this feature, go to the following articles:

- How to gain access to local files in a remote desktop session to a Windows XP-based or to a Windows Server 2003-based host computer
- Make Local Devices and Resources Available in a Remote Session
- Getting Started with Remote Desktop Client on Mac

Stop and Start Your Instance

You can stop and restart your instance if it has an Amazon EBS volume as its root device. The instance retains its instance ID, but can change as described in the Overview section.

When you stop an instance, we shut it down. We don't charge hourly usage for a stopped instance, or data transfer fees, but we do charge for the storage for any Amazon EBS volumes. Each time you start a stopped instance we charge a full instance hour, even if you make this transition multiple times within a single hour.

While the instance is stopped, you can treat its root volume like any other volume, and modify it (for example, repair file system problems or update software). You just detach the volume from the stopped

instance, attach it to a running instance, make your changes, detach it from the running instance, and then reattach it to the stopped instance. Make sure that you reattach it using the storage device name that's specified as the root device in the block device mapping for the instance.

If you decide that you no longer need an instance, you can terminate it. As soon as the state of an instance changes to `shutting-down` or `terminated`, we stop charging for that instance. For more information, see Terminate Your Instance (p. 224).

Contents

- Overview (p. 219)
- Stopping and Starting Your Instances (p. 220)
- Modifying a Stopped Instance (p. 221)
- Troubleshooting (p. 221)

Overview

You can only stop an Amazon EBS-backed instance. To verify the root device type of your instance, describe the instance and check whether the device type of its root volume is `ebs` (Amazon EBS-backed instance) or `instance store` (instance store-backed instance). For more information, see Determining the Root Device Type of Your AMI (p. 54).

When you stop a running instance, the following happens:

- The instance performs a normal shutdown and stops running; its status changes to `stopping` and then `stopped`.
- Any Amazon EBS volumes remain attached to the instance, and their data persists.
- Any data stored in the RAM of the host computer or the instance store volumes of the host computer is gone.
- EC2-Classic: We release the public and private IP addresses for the instance when you stop the instance, and assign new ones when you restart it.

  EC2-VPC: The instance retains its private IP addresses when stopped and restarted. We release the public IP address and assign a new one when you restart it.
- EC2-Classic: We disassociate any Elastic IP address that's associated with the instance. You're charged for Elastic IP addresses that aren't associated with an instance. When you restart the instance, you must associate the Elastic IP address with the instance; we don't do this automatically.

  EC2-VPC: The instance retains its associated Elastic IP addresses. You're charged for any Elastic IP addresses associated with a stopped instance.
- When you stop and restart a Windows instance, by default, we change the instance host name to match the new IP address and initiate a reboot. By default, we also change the drive letters for any attached Amazon EBS volumes. For more information about these defaults and how you can change them, see Configuring a Windows Instance Using the EC2Config Service (p. 235) in the *Amazon EC2 User Guide for Microsoft Windows Instances*.
- If you've registered the instance with a load balancer, it's likely that the load balancer won't be able to route traffic to your instance after you've stopped and restarted it. You must de-register the instance from the load balancer after stopping the instance, and then re-register after starting the instance. For more information, see De-Register and Register EC2 Instances with Your Load Balancer in the *Elastic Load Balancing Developer Guide*.
- If your instance is in an Auto Scaling group, the Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance. For more information, see Health Checks for Auto Scaling Instances in the *Auto Scaling Developer Guide*.

- When you stop a ClassicLink instance, it's unlinked from the VPC to which it was linked. You must link the instance to the VPC again after restarting it. For more information about ClassicLink, see ClassicLink (p. 457).

For more information, see Differences Between Reboot, Stop, and Terminate (p. 205).

You can modify the following attributes of an instance only when it is stopped:

- Instance type
- User data
- Kernel
- RAM disk

If you try to modify these attributes while the instance is running, Amazon EC2 returns the `IncorrectInstanceState` error.

Stopping and Starting Your Instances

You can start and stop your Amazon EBS-backed instance using the console or the command line.

By default, when you initiate a shutdown from an Amazon EBS-backed instance (using the **shutdown**, **halt**, or **poweroff** command), the instance stops. You can change this behavior so that it terminates instead. For more information, see Changing the Instance Initiated Shutdown Behavior (p. 227).

To stop and start an Amazon EBS-backed instance using the console

1. In the navigation pane, choose **Instances**, and select the instance.
2. [EC2-Classic] If the instance has an associated Elastic IP address, write down the Elastic IP address and the instance ID shown in the details pane.
3. Choose **Actions**, select **Instance State**, and then choose **Stop**. If **Stop** is disabled, either the instance is already stopped or its root device is an instance store volume.

   Warning
   When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.
4. In the confirmation dialog box, choose **Yes, Stop**. It can take a few minutes for the instance to stop.

   [EC2-Classic] When the instance state becomes `stopped`, the **Elastic IP**, **Public DNS**, **Private DNS**, and **Private IPs** fields in the details pane are blank to indicate that the old values are no longer associated with the instance.
5. While your instance is stopped, you can modify certain instance attributes. For more information, see Modifying a Stopped Instance (p. 221).
6. To restart the stopped instance, select the instance, choose **Actions**, select **Instance State**, and then choose **Start**.
7. In the confirmation dialog box, choose **Yes, Start**. It can take a few minutes for the instance to enter the `running` state.

   [EC2-Classic] When the instance state becomes `running`, the **Public DNS**, **Private DNS**, and **Private IPs** fields in the details pane contain the new values that we assigned to the instance.
8. [EC2-Classic] If your instance had an associated Elastic IP address, you must reassociate it as follows:

   a. In the navigation pane, choose **Elastic IPs**.

b. Select the Elastic IP address that you wrote down before you stopped the instance.
c. Choose **Actions**, and then select **Associate Address**.
d. Select the instance ID that you wrote down before you stopped the instance, and then choose **Associate**.

To stop and start an Amazon EBS-backed instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- stop-instances and start-instances (AWS CLI)
- ec2-stop-instances and ec2-start-instances (Amazon EC2 CLI)
- Stop-EC2Instance and Start-EC2Instance (AWS Tools for Windows PowerShell)

Modifying a Stopped Instance

You can change the instance type, user data, and EBS-optimization attributes of a stopped instance using the AWS Management Console or the command line interface. You can't use the AWS Management Console to modify the kernel or RAM disk attributes.

To change the instance type for a stopped instance using the console

For information about the limitations, and step-by-step directions, see Resizing Your Instance (p. 118).

To change the user data for a stopped instance using the console

For information about the limitations, and step-by-step directions, see Adding User Data (p. 163).

To enable or disable EBS–optimization for your instance using the console

For more information and step-by-step directions, see Modifying EBS–Optimization (p. 557).

To modify an instance attribute using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- modify-instance-attribute (AWS CLI)
- ec2-modify-instance-attribute (Amazon EC2 CLI)
- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

Troubleshooting

If you have stopped your Amazon EBS-backed instance and it appears "stuck" in the `stopping` state, you can forcibly stop it. For more information, see Troubleshooting Stopping Your Instance in the *Amazon EC2 User Guide for Linux Instances*.

Reboot Your Instance

An instance reboot is equivalent to an operating system reboot. In most cases, it takes only a few minutes to reboot your instance. When you reboot an instance, it remains on the same physical host, so your instance keeps its public DNS name, private IP address, and any data on its instance store volumes.

Rebooting an instance doesn't start a new instance billing hour, unlike stopping and restarting your instance.

We might schedule your instance for a reboot for necessary maintenance, such as to apply updates that require a reboot. No action is required on your part; we recommend that you wait for the reboot to occur within its scheduled window. For more information, see Scheduled Events for Your Instances (p. 326).

We recommend that you use Amazon EC2 to reboot your instance instead of running the operating system reboot command from your instance. If you use Amazon EC2 to reboot your instance, we perform a hard reboot if the instance does not cleanly shut down within four minutes. If you use AWS CloudTrail, then using Amazon EC2 to reboot your instance also creates an API record of when your instance was rebooted.

To reboot an instance using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**.
3. Select the instance, click **Actions**, select **Instance State**, and then click **Reboot**.
4. Click **Yes, Reboot** when prompted for confirmation.

To reboot an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- reboot-instances (AWS CLI)
- ec2-reboot-instances (Amazon EC2 CLI)
- Restart-EC2Instance (AWS Tools for Windows PowerShell)

Instance Retirement

An instance is scheduled to be retired when AWS detects irreparable failure of the underlying hardware hosting the instance. When an instance reaches its scheduled retirement date, it is stopped or terminated by AWS. If your instance root device is an Amazon EBS volume, the instance is stopped, and you can start it again at any time. Starting the stopped instance migrates it to new hardware. If your instance root device is an instance store volume, the instance is terminated, and cannot be used again.

Topics

- Identifying Instances Scheduled for Retirement (p. 222)
- Working with Instances Scheduled for Retirement (p. 223)

For more information about types of instance events, see Scheduled Events for Your Instances (p. 326).

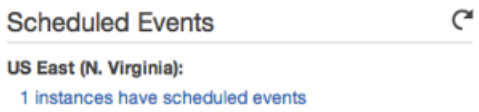
Identifying Instances Scheduled for Retirement

If your instance is scheduled for retirement, you'll receive an email prior to the event with the instance ID and retirement date. This email is sent to the address that's associated with your account; the same email

address that you use to log in to the AWS Management Console. If you use an email account that you do not check regularly, then you can use the Amazon EC2 console or the command line to determine if any of your instances are scheduled for retirement. To update the contact information for your account, go to the Account Settings page.

To identify instances scheduled for retirement using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **EC2 Dashboard**. Under **Scheduled Events**, you can see the events associated with your Amazon EC2 instances and volumes, organized by region.



3. If you have an instance with a scheduled event listed, click its link below the region name to go to the **Events** page.
4. The **Events** page lists all resources with events associated with them. To view instances that are scheduled for retirement, select **Instance resources** from the first filter list, and then **Instance retirement** from the second filter list.
5. If the filter results show that an instance is scheduled for retirement, select it, and note the date and time in the **Start time** field in the details pane. This is your instance retirement date.

To identify instances scheduled for retirement using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-instance-status (AWS CLI)
- ec2-describe-instance-status (Amazon EC2 CLI)
- Get-EC2InstanceStatus (AWS Tools for Windows PowerShell)

Working with Instances Scheduled for Retirement

There are a number of actions available to you when your instance is scheduled for retirement. The action you take depends on whether your instance root device is an Amazon EBS volume, or an instance store volume. If you do not know what your instance root device type is, you can find out using the Amazon EC2 console or the command line.

Determining Your Instance Root Device Type

To determine your instance root device type using the console

1. In the navigation pane, click **Events**. Use the filter lists to identify retiring instances, as demonstrated in the procedure above, Identifying instances scheduled for retirement (p. 223).
2. In the **Resource ID** column, click the instance ID to go to the **Instances** page.
3. Select the instance and locate the **Root device type** field in the **Description** tab. If the value is `ebs`, then your instance is EBS-backed. If the value is `instance-store`, then your instance is instance store-backed.

To determine your instance root device type using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-instances (AWS CLI)
- ec2-describe-instances (Amazon EC2 CLI)
- Get-EC2Instance (AWS Tools for Windows PowerShell)

Managing Instances Scheduled for Retirement

You can perform one of the actions listed below in order to preserve the data on your retiring instance. It's important that you take this action before the instance retirement date, to prevent unforeseen downtime and data loss.

> **Warning**
> If your instance store-backed instance passes its retirement date, it's terminated and you cannot recover the instance or any data that was stored on it. Regardless of the root device of your instance, the data on instance store volumes is lost when the instance is retired, even if they are attached to an EBS-backed instance.

Instance Root Device Type	Action
EBS	Wait for the scheduled retirement date - when the instance is stopped - or stop the instance yourself before the retirement date. You can start the instance again at any time. For more information about stopping and starting your instance, and what to expect when your instance is stopped, such as the effect on public, private and Elastic IP addresses associated with your instance, see Stop and Start Your Instance (p. 218).
EBS	Create an EBS-backed AMI from your instance, and launch a replacement instance. For more information, see Creating an Amazon EBS-Backed Windows AMI (p. 68).
Instance store	Bundle your instance, and then create an instance store-backed AMI from the manifest that's created during bundling. You can launch a replacement instance from your new AMI. For more information, see Creating an Instance Store-Backed Windows AMI (p. 70).

Terminate Your Instance

When you've decided that you no longer need an instance, you can terminate it. As soon as the state of an instance changes to `shutting-down` or `terminated`, you stop incurring charges for that instance.

You can't connect to or restart an instance after you've terminated it. However, you can launch additional instances using the same AMI. If you'd rather stop and restart your instance, see Stop and Start Your Instance (p. 218). For more information, see Differences Between Reboot, Stop, and Terminate (p. 205).

Topics

- Instance Termination (p. 225)
- Terminating an Instance (p. 225)
- Enabling Termination Protection for an Instance (p. 226)
- Changing the Instance Initiated Shutdown Behavior (p. 227)

- Preserving Amazon EBS Volumes on Instance Termination (p. 227)

Instance Termination

After you terminate an instance, it remains visible in the console for a short while, and then the entry is deleted.

When an instance terminates, the data on any instance store volumes associated with that instance is deleted.

By default, Amazon EBS root device volumes are automatically deleted when the instance terminates. However, by default, any additional EBS volumes that you attach at launch, or any EBS volumes that you attach to an existing instance persist even after the instance terminates. This behavior is controlled by the volume's `DeleteOnTermination` attribute, which you can modify. For more information, see Preserving Amazon EBS Volumes on Instance Termination (p. 227).

You can prevent an instance from being terminated accidentally by someone using the AWS Management Console, the CLI, and the API. This feature is available for both Amazon EC2 instance store-backed and Amazon EBS-backed instances. Each instance has a `DisableApiTermination` attribute with the default value of `false` (the instance can be terminated through Amazon EC2). You can modify this instance attribute while the instance is running or stopped (in the case of Amazon EBS-backed instances). For more information, see Enabling Termination Protection for an Instance (p. 226).

You can control whether an instance should stop or terminate when shutdown is initiated from the instance using an operating system command for system shutdown. For more information, see Changing the Instance Initiated Shutdown Behavior (p. 227).

If you run a script on instance termination, your instance might have an abnormal termination, because we have no way to ensure that shutdown scripts run. Amazon EC2 attempts to shut an instance down cleanly and run any system shutdown scripts; however, certain events (such as hardware failure) may prevent these system shutdown scripts from running.

Terminating an Instance

You can terminate an instance using the AWS Management Console or the command line.

To terminate an instance using the console

1. Before you terminate the instance, verify that you won't lose any data by checking that your Amazon EBS volumes won't be deleted on termination and that you've copied any data that you need from your instance store volumes to Amazon EBS or Amazon S3.
2. Open the Amazon EC2 console.
3. In the navigation pane, click **Instances**.
4. Select the instance, click **Actions**, select **Instance State**, and then click **Terminate**.
5. Click **Yes, Terminate** when prompted for confirmation.

To terminate an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- terminate-instances (AWS CLI)
- ec2-terminate-instances (Amazon EC2 CLI)
- Stop-EC2Instance (AWS Tools for Windows PowerShell)

Enabling Termination Protection for an Instance

By default, you can terminate your instance using the Amazon EC2 console, command line interface, or API. If you want to prevent your instance from being accidentally terminated using Amazon EC2, you can enable *termination protection* for the instance. The `DisableApiTermination` attribute controls whether the instance can be terminated using the console, CLI, or API. By default, termination protection is disabled for your instance. You can set the value of this attribute when you launch the instance, while the instance is running, or while the instance is stopped (for Amazon EBS-backed instances).

The `DisableApiTermination` attribute does not prevent you from terminating an instance by initiating shutdown from the instance (using an operating system command for system shutdown) when the `InstanceInitiatedShutdownBehavior` attribute is set. For more information, see Changing the Instance Initiated Shutdown Behavior (p. 227).

You can't prevent instances that are part of an Auto Scaling group from terminating using termination protection. However, you can specify which instances should terminate first. For more information, see Choosing a Termination Policy in the *Auto Scaling Developer Guide*.

You can't enable termination protection for Spot instances — a Spot instance is terminated when the Spot price exceeds your bid price. However, you can prepare your application to handle Spot instance interruptions. For more information, see Spot Instance Interruptions (p. 155).

You can enable or disable termination protection using the AWS Management Console or the command line.

To enable termination protection for an instance at launch time

1. On the dashboard of the Amazon EC2 console, click **Launch Instance** and follow the directions in the wizard.
2. On the **Configure Instance Details** page, select the **Enable termination protection** check box.

To enable termination protection for a running or stopped instance

1. Select the instance, click **Actions**, and then click **Change Termination Protection**.
2. Click **Yes, Enable**.

To disable termination protection for a running or stopped instance

1. Select the instance, click **Actions**, select **Instance Settings**, and then click **Change Termination Protection**.
2. Click **Yes, Disable**.

To enable or disable termination protection using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- modify-instance-attribute (AWS CLI)
- ec2-modify-instance-attribute (Amazon EC2 CLI)
- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

Changing the Instance Initiated Shutdown Behavior

By default, when you initiate a shutdown from an Amazon EBS-backed instance (using a command such as **shutdown**, **halt**, or **poweroff**), the instance stops. You can change this behavior using the `InstanceInitiatedShutdownBehavior` attribute for the instance so that it terminates instead. You can update this attribute while the instance is running or stopped.

You can update the `InstanceInitiatedShutdownBehavior` attribute using the AWS Management Console or the command line.

To change the shutdown behavior of an instance using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**.
3. Select the instance, click **Actions**, select **Instance Settings**, and then click **Change Shutdown Behavior**. The current behavior is already selected.
4. To change the behavior, select an option from the **Shutdown behavior** list, and then click **Apply**.

To change the shutdown behavior of an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- modify-instance-attribute (AWS CLI)
- ec2-modify-instance-attribute (Amazon EC2 CLI)
- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

Preserving Amazon EBS Volumes on Instance Termination

By default, we do the following:

- Preserve any non-root device volumes that you attach at launch, and any volumes that you attach to an existing instance, even after the instance terminates
- Preserve any attached EBS volumes when you stop and restart an instance
- Delete the root device volume when you terminate the instance

You can change this behavior using the `DeleteOnTermination` attribute for the volume. If the value of this attribute is `true`, we delete the volume after the instance terminates; if the `DeleteOnTermination` attribute is `false`, the volume persists in its current state. You can take a snapshot of the volume, and you can attach it to another instance.

Any time you attach an EBS volume to an existing instance, its `DeleteOnTermination` attribute is set to `false`.

You can see the value for the `DeleteOnTermination` attribute on the volumes attached to an instance by looking at the instance's block device mapping. For more information, see Viewing the EBS Volumes in an Instance Block Device Mapping (p. 595).

You can update the `DeleteOnTermination` attribute using the AWS Management Console or the command line.

Changing the Root Volume to Persist Using the Console

Using the console, you can change the `DeleteOnTermination` attribute when you launch an instance. To change this attribute for a running instance, you must use the command line.

To change the root volume of an instance to persist at launch using the console

1. Open the Amazon EC2 console.
2. From the console dashboard, click **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, choose an AMI and click **Select**.
4. Follow the wizard to complete the **Choose an Instance Type** and **Configure Instance Details** pages.
5. On the **Add Storage** page, deselect the **Delete On Termination** check box for the root volume.
6. Complete the remaining wizard pages, and then click **Launch**.

You can verify the setting by viewing details for the root device volume on the instance's details pane. Next to **Block devices**, click the entry for the root device volume. By default, **Delete on termination** is `True`. If you change the default behavior, **Delete on termination** is `False`.

Changing the Root Volume of a Running Instance to Persist Using the Command Line

You can use one of the following commands to change the root device volume of a running instance to persist. The root device is typically `xvda`. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- modify-instance-attribute (AWS CLI)
- ec2-modify-instance-attribute (Amazon EC2 CLI)
- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

Example for AWS CLI

The following command preserves the root volume by setting its `DeleteOnTermination` attributes to `false`.

```
C:\> aws ec2 modify-instance-attribute --instance-id i-5203422c --block-device-
mappings  "[{\"DeviceName\":\"xvda\",\"Ebs\":{\"DeleteOnTermination\":false}}]"
```

You can confirm that `deleteOnTermination` is `false` by using the describe-instances command and looking for the `BlockDeviceMappings` entry for `xvda` in the command output.

Example for Amazon EC2 CLI

The following command preserves the root volume by setting its `DeleteOnTermination` attribute to `false`.

```
C:\> ec2-modify-instance-attribute i-5203422c -b "xvda=::false"
```

Changing the Root Volume of an Instance to Persist at Launch Using the Command Line

When you launch an instance, you can use one of the following commands to change the root device volume to persist. The root device is typically `xvda`. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- run-instances (AWS CLI)
- ec2-run-instances (Amazon EC2 CLI)
- New-EC2Instance (AWS Tools for Windows PowerShell)

Example for AWS CLI

The following command preserves the root volume by setting its `DeleteOnTermination` attributes to `false`.

```
C:\> aws ec2 run-instances --image-id ami-1a2b3c4d --block-device-mappings
"[{\"DeviceName\":\"xvda\",\"Ebs\":{\"DeleteOnTermination\":false}}]" other
parameters...
```

You can confirm that `deleteOnTermination` is `false` by using the describe-instances command and looking for the `BlockDeviceMappings` entry for `xvda` in the command output.

Example for Amazon EC2 CLI

The following command preserves the root volume by setting its `DeleteOnTermination` attribute to `false`.

```
C:\> ec2-run-instances ami-1a2b3c4d  -b "xvda=::false" other parameters... -v
```

Recover Your Instance

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair. A recovered instance is identical to the original instance, including the instance ID, private IP addresses, Elastic IP addresses, and all instance metadata. For more information about using Amazon CloudWatch alarms to recover an instance, see Create Alarms That Stop, Terminate, or Recover an Instance in the *Amazon EC2 User Guide for Linux Instances*. To troubleshoot issues with instance recovery failures, see Troubleshooting Instance Recovery Failures in the *Amazon EC2 User Guide for Linux Instances*.

When the `StatusCheckFailed_System` alarm is triggered, and the recover action is initiated, you will be notified by the Amazon SNS topic that you selected when you created the alarm and associated the recover action. During instance recovery, the instance is migrated during an instance reboot, and any data that is in-memory is lost. When the process is complete, you'll receive an email notification that includes the status of the recovery attempt and any further instructions. You will notice an instance reboot on the recovered instance.

Examples of problems that cause system status checks to fail include:

- Loss of network connectivity
- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host

Important
The recover action is only supported on:

- C3, C4, M3, R3, and T2 instance types.
- Instances in the Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), EU (Ireland), EU (Frankfurt), South America (Sao Paulo), US East (N. Virginia), US West (N. California) and US West (Oregon) regions.
- Instances in a VPC.

  Note
  If your instance has a public IP address, it receives a new public IP address after recovery (if your subnet setting allows it). To retain the public IP address, use an Elastic IP address instead.
- Instances with shared tenancy (where the tenancy attribute of the instance is set to `default`).
- Instances that use Amazon EBS storage exclusively.

Currently, the recover action is not supported for EC2-Classic instances, dedicated tenancy instances, and instances that use any instance store volumes, including instances launched with block device mappings for instance store volumes.

Note
If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions:

- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` for all alarms on Amazon EC2 instance status metrics.
- `ec2:StopInstances` for alarms with stop actions.
- `ec2:TerminateInstances` for alarms with terminate actions.
- `ec2:DescribeInstanceRecoveryAttribute`, and `ec2:RecoverInstances` for alarms with recover actions.

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see Permissions and Policies in the *IAM User Guide*.
If you are using an IAM role (e.g., an Amazon EC2 instance profile), you cannot stop or terminate the instance using alarm actions. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.

If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot stop or terminate an Amazon EC2 instance using alarm actions.

Upgrading a Windows Server EC2 Instance to a Newer Version of Windows Server

This topic steps you through the process of upgrading a Windows Server EC2 instance to a new version of Windows Server. You might choose to upgrade because of feature and security enhancements in a newer operating system or because Microsoft officially stops supporting the Windows Server 2003 operating system on July 14, 2015. Upgrading an EC2 instance to a newer version of Windows Server can be complicated by incompatible drivers and applications. These incompatibilities can cause a wide range of problems, including upgrade failures, system failures, and loss of network connectivity. This topic includes steps to help mitigate these types of errors or failures.

There are two methods for porting an older version of Windows Server to a newer version: migration and upgrade. This topic covers the upgrade method and focuses on a known issue during the upgrade process where Setup removes portions of the para-virtual (PV) drivers that enable a user to connect to the instance by using Remote Desktop. This topic also briefly describes the migration method because Microsoft has traditionally recommended migrating to a newer version of Windows Server instead of upgrading. Migrating can result in fewer upgrade errors or issues, but can take longer than an in-place upgrade.

Migration

Migrating involves capturing settings, configurations, and data and porting these to a newer operating system on separate hardware. Once validated, the migrated system can be promoted to production. You can migrate EC2 instances by launching a new instance from an AMI of the new operating system. You can streamline the process further by using AWS CloudFormation and Amazon EC2 Simple Systems Manager to automatically apply settings and configurations to the new system with little manual work.

To migrate your server

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **AMIs**.
3. Choose **Owned by me**, and then choose **Public images**.
4. In the **Search** field, add the following filters and press Enter.

    a. Owner : Amazon images
    b. AMI Name : Windows_Server-2008

    Note
    The **Search** field is case sensitive.

5. Launch a new instance from an AMI.
6. Log onto the new instance and install all updates.
7. Perform application installation and configuration changes.
8. Test the server.
9. When validated, promote the server to production.

Upgrade

PV drivers running on Windows Server EC2 instances enable you to access the instance by using Remote Desktop. During an operating system upgrade, Setup removes portions of these drivers, which causes

the instance to be unreachable after the upgrade. This section describes the different phases of the upgrade process with the Upgrade Helper Service, the steps for running the service, and how to troubleshoot issues.

> **Important**
> AWS provides upgrade support for issues or problems with the Upgrade Helper Service. For all other issues or problems with an operating system upgrade or migration we recommend reviewing the TechNet articles listed in the *Before You Begin* section of this document.

About the Upgrade Helper Service

You must run UpgradeHelperService.exe before you start the upgrade. After you run it, the utility creates a Windows service that executes during the post-upgrade steps to correct the driver state. The executable is written in C# and can run on .NET Framework versions 2.0 through 4.0.

When you run UpgradeHelperService.exe on the system *before* the upgrade it performs the following tasks:

- Creates a new Windows service called *UpgradeHelperService*.
- Verifies that Citrix PV drivers are installed.
- Checks for unsigned boot critical drivers and presents a warning if any are found. Unsigned boot critical drivers could cause system failure after the upgrade if the drivers are not compatible with the newer Windows Server version.

When you run UpgradeHelperService.exe on the system *after* the upgrade it performs the following tasks:

- Enables the RealTimeIsUniversal registry key for correct time synchronization in Amazon Elastic Compute Cloud (Amazon EC2).
- Restores the missing PV driver by executing the following command:

  pnputil -i -a "C:\Program Files (x86)\Citrix\XenTools*.inf"
- Installs the missing device by executing the following command:

  C:\Temp\EC2DriverUtils.exe install "C:\Program Files (x86)\Citrix\XenTools\xevtchn.inf" ROOT\XENEVTCHN
- Once complete, automatically removes the *UpgradeHelperService* Windows service.

Before You Begin

Complete the following tasks and note the following important details before you upgrade.

- Read the Microsoft documentation to understand the upgrade requirements, known issues, and restrictions. You should also review the official instructions for upgrading. The Upgrade Helper Service does not fix operating system upgrade issues. The service only fixes problems with Amazon EC2 connectivity. For more information, see the following topics on Microsoft TechNet.
  - Upgrading to Windows Server 2008
  - Upgrading to Windows Server 2008 R2
  - Upgrading to Windows Server 2012
  - Upgrading to Windows Server 2012 R2
- The Upgrade Helper Service only supports instances running Citrix PV drivers. If the instance is running Red Hat drivers, you must manually upgrade those drivers before you upgrade.
- We do not recommend performing an operating system upgrade on a T1 or T2 instance type. These types of instances might not have enough resources to manage the upgrade process. If you need to upgrade one of these instances, you must resize the instance to another instance type, perform the upgrade, and then resize it back to a T1 or T2 instance type.

- Create an AMI of the system you plan to upgrade for either backup or testing purposes. You can then perform the upgrade on the copy to simulate a test environment. If the upgrade completes, you can switch traffic to this instance with little downtime. If the upgrade fails, you can revert to the backup.
- Verify that the root volume on your Windows instance has enough free disk space. The Windows Setup process might not warn you of insufficient disk space. For information about how much disk space is required to upgrade a specific operating system, see the Microsoft documentation. If the volume does not have enough space, it can be expanded. For more information, see Expanding the Storage Space of an EBS Volume on Windows.
- Determine your upgrade path. You must upgrade the operating system to the same architecture. For example, you must upgrade a 32-bit system to a 32-bit system. Windows Server 2008 R2 and later are 64-bit only.

Performing the Upgrade

This procedure describes how to attach the installation media volume to your EC2 instance and how to upgrade the instance by using UpgradeHelperService.exe.

To upgrade an EC2 Windows Server instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Locate the instance ID and Availability Zone for the Windows Server EC2 instance that you want to upgrade. You will need to specify this information when you create and attach the Windows installation media volume later in this procedure.
3. Create a new volume from a Windows Server installation media snapshot.
    a. In the EC2 console, choose **Snapshots**.
    b. Choose **Owned by me** and then choose **Public Snapshots**.
    c. In the **Search** field, add the following filters and press Enter.
        i. Owner : Amazon images
        ii. Description : Windows
    d. Select the snapshot that matches your system architecture. For example, select **Windows 2008 64-bit Installation Media** if your Windows Server 2003 instance is 64-bit.
    e. From the context menu choose (right-click) **Create Volume**.
    f. In the **Create Volume** dialog box, choose the Availability Zone that matches your Windows instance, and then choose **Create**.
4. In the **Volume Successfully Created** message, choose the volume you just created.
5. Select the volume in the list and then choose (right-click) **Attach Volume** from the context menu.
6. In the **Attach Volume** dialog box, type the instance ID, and choose **Attach**.
7. On your Windows instance, on the C:\ drive, create a new folder called temp. This folder must be available in the same location after the upgrade. Creating the temp folder in a Windows system folder or a user profile folder, such as the desktop, can cause the upgrade to fail.
8. Download OSUpgrade.zip and extract the files into the C:\temp folder.
9. Run UpgradeHelperService.exe from c:\temp and review the Log.txt file in c:\temp for any warnings.
10. Use Microsoft Knowledge Base article 950376 to uninstall PowerShell from a Windows 2003 instance, or perform the following unsupported steps to bypass the Windows Upgrade check:
    a. In Windows Explorer, choose **WINDOWS**, and then choose **System32**.
    b. Rename the WindowsPowerShell folder to *old*WindowsPowerShell. For 64-bit instances, you must also rename the WindowsPowerShell folder in the **WINDOWS** > **SysWow64** folder.

11. Begin the upgrade by using Windows Explorer to open the Installation Media volume you attached to the instance earlier in this procedure.
12. In the **Sources** folder, run Setup.exe.
13. On the **Select the operating system you want to install** page, select the *Full Installation* SKU that matches your Windows Server instance, and choose **Next**.
14. On the **Which type of installation do you want?** page, choose **Upgrade**.
15. Complete the Setup wizard.

Windows Server Setup will then copy and process files. After several minutes, your Remote Desktop session closes. The time it takes to upgrade will depend on the number of applications and server roles running on your Windows Server instance. The upgrade process could take as little as 40 minutes or as long as several hours. The instance will fail status check 1 of 2 in the EC2 console during the upgrade process. When the upgrade completes, both status checks pass. You can check the system log for console outputs or refer to Amazon CloudWatch monitors for disk and CPU activity to determine if the upgrade is not progressing.

If the instance has not passed both status checks after several hours see *Troubleshooting the Upgrade* in this topic.

Post Upgrade Tasks

1. Log into the instance to initiate an upgrade for the .NET Framework and reboot the system when prompted.
2. Install the latest version of the EC2Config service.
3. Install Microsoft hotfix KB2800213.
4. Install Microsoft hotfix KB2922223.
5. If you upgraded to Windows Server 2012 R2, we recommend that you upgrade the PV drivers to AWS PV drivers when they are available. For more information, see Important information about Amazon EC2 instances running Windows Server 2012 R2 .

Troubleshooting the Upgrade

This section can help you locate and diagnose errors or failures

- If the instance has not passed both status checks after several hours do the following.
  - If you upgraded to Windows Server 2008 and both status checks fail after several hours, the upgrade may have failed and be presenting a prompt to **Click OK** to confirm rolling back. Because the console is not accessible at this state, there is no way to click the button. To get around this, perform a reboot via the EC2 console or API. The reboot will take ten minutes or more to initiate. The instance might become available after 25 minutes.
  - Remove applications or server roles from the server and try again.
- If the instance does not pass both status checks after removing applications or server roles from the server, do the following.
  - Stop the instance and attach the root volume to another instance. For more information, see the description of how to stop and attach the root volume to another instance in Waiting for the metadata service.
  - Analyze Windows Setup log files and event logs for failures.

Configuring Your Windows Instance

A Windows instance is a virtual server running Microsoft Windows Server in the cloud.

After you have successfully launched and logged into your instance, you can make changes to it so that it's configured to meet the needs of a specific application. The following are some common tasks to help you get started.

Tasks

- Configuring a Windows Instance Using the EC2Config Service (p. 235)
- Paravirtual Drivers (p. 258)
- Setting Passwords for Windows Instances (p. 277)
- Setting the Time for a Windows Instance (p. 281)
- Managing Windows Instance Configuration (p. 284)
- Joining a Windows Instance to an AWS Directory Service Domain (p. 291)
- Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs (p. 299)
- Configuring a Secondary Private IP Address for Your Windows Instance in a VPC (p. 312)

Configuring a Windows Instance Using the EC2Config Service

AWS Windows AMIs contain an additional service installed by Amazon Web Services, the EC2Config service. Although optional, this service provides access to advanced features that aren't otherwise available. This service runs in the LocalSystem account and performs tasks on the instance. For example, it can send Windows event logs and IIS request logs to Amazon CloudWatch Logs. For more information about how to configure EC2Config for use with CloudWatch Logs, see Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs (p. 245). The service binaries and additional files are contained in the `%ProgramFiles%\Amazon\EC2ConfigService` directory.

The EC2Config service is started when the instance is booted. It performs tasks during initial instance startup and each time you stop and start the instance. It can also perform tasks on demand. Some of these tasks are automatically enabled, while others must be enabled manually. EC2Config uses settings

files to control its operation. You can update these settings files using either a graphical tool or by directly editing XML files.

The EC2Config service runs Sysprep, a Microsoft tool that enables you to create a customized Windows AMI that can be reused. For more information, see Create a Standard Amazon Machine Image Using Sysprep (p. 90).

When EC2Config calls Sysprep, it uses the settings files in `EC2ConfigService\Settings` to determine which operations to perform. You can edit these files indirectly using the **Ec2 Service Properties** dialog box, or directly using an XML editor or a text editor. However, there are some advanced settings that aren't available in the **Ec2 Service Properties** dialog box, so you must edit those entries directly.

If you create an AMI from an instance after updating its settings, the new settings are applied to any instance that's launched from the new AMI. For information about creating an AMI, see Creating an Amazon EBS-Backed Windows AMI (p. 68).

Contents

- Overview of EC2Config Tasks (p. 236)
- Ec2 Service Properties (p. 237)
- EC2Config Settings Files (p. 240)
- Executing User Data (p. 243)
- Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs (p. 245)
- Installing the Latest Version of EC2Config (p. 256)
- Stopping, Restarting, Deleting, or Uninstalling EC2Config (p. 258)

Overview of EC2Config Tasks

EC2Config runs initial startup tasks when the instance is first started and then disables them. To run these tasks again, you must explicitly enable them prior to shutting down the instance, or by running Sysprep manually. These tasks are as follows:

- Set a random, encrypted password for the administrator account.
- Generate and install the host certificate used for Remote Desktop Connection.
- Dynamically extend the operating system partition to include any unpartitioned space.
- Execute the specified user data (and Cloud-Init, if it's installed).

EC2Config performs the following tasks every time the instance starts:

- Change the host name to match the private IP address in Hex notation (this task is disabled by default and must be enabled in order to run at instance start).
- Configure the key management server (KMS), check for Windows activation status, and activate Windows as necessary.
- Format and mount all Amazon EBS volumes and instance store volumes, and map volume names to drive letters.
- Write event log entries to the console to help with troubleshooting (this task is disabled by default and must be enabled in order to run at instance start).
- Write to the console that Windows is ready.
- Add a custom route to the primary network adapter to enable the following IP addresses when multiple NICs are attached: 169.254.169.250, 169.254.169.251, and 169.254.169.254. These addresses are used by Windows Activation and when you access instance metadata.

EC2Config performs the following task every time a user logs in:

- Display wallpaper information to the desktop background.

While the instance is running, you can request that EC2Config perform the following task on demand:

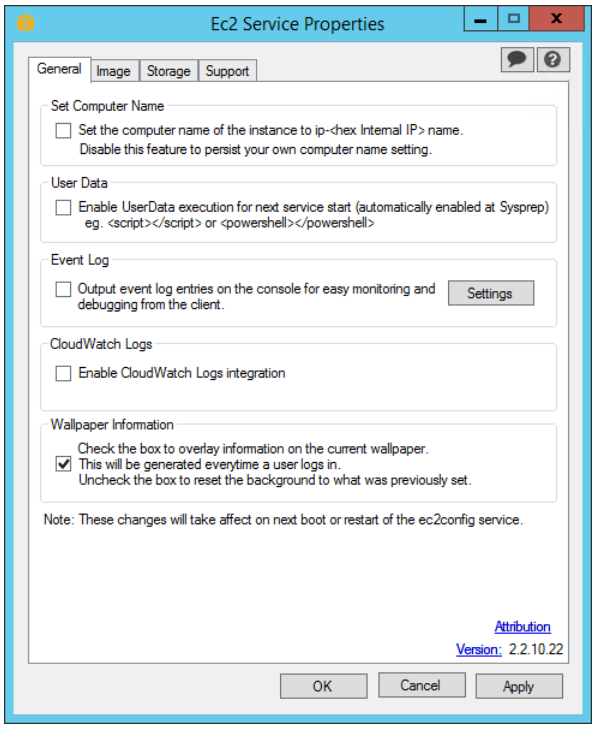
- Run Sysprep and shut down the instance so that you can create an AMI from it. For more information, see Create a Standard Amazon Machine Image Using Sysprep (p. 90).

Ec2 Service Properties

The following procedure describes how to use the **Ec2 Service Properties** dialog box to enable or disable settings.

To change settings using the Ec2 Service Properties dialog box

1. Launch and connect to your Windows instance.
2. From the **Start** menu, click **All Programs**, and then click **EC2ConfigService Settings**.



3. On the **General** tab of the **Ec2 Service Properties** dialog box, you can enable or disable the following settings.

   Set Computer Name
   If this setting is enabled (it is disabled by default), the host name is compared to the current internal IP address at each boot; if the host name and internal IP address do not match, the host name is reset to contain the internal IP address and then the system reboots to pick up the new host name. To set your own host name, or to prevent your existing host name from being modified, do not enable this setting.

   User Data
   User data execution enables you to inject scripts into the instance metadata during the first launch. From an instance, you can read user data at http://169.254.169.254/latest/user-data/.

This information remains static for the life of the instance, persisting when the instance is stopped and started, until it is terminated.

If you use a large script, we recommend that you use user data to download the script, and then execute it.

For more information, see Executing User Data (p. 243).

Event Log

Use this setting to display event log entries on the console during boot for easy monitoring and debugging.

Click **Settings** to specify filters for the log entries sent to the console. The default filter sends the three most recent error entries from the system event log to the console.

CloudWatch Logs

Starting with EC2Config version 2.2.5 (version 2.2.6 or later is recommended), you can export all Windows Server messages in the System log, Security log, Application log, and IIS log to CloudWatch Logs and monitor them using CloudWatch metrics. EC2Config version 2.2.10 or later adds the ability to export any event log data, Event Tracing (Windows) data, or text-based log files to CloudWatch Logs. In addition, you can also export performance counter data to CloudWatch For more information, see Monitoring System, Application, and Custom Log Files in the Amazon CloudWatch Developer Guide.

1. Select **Enable CloudWatch integration**, and then click **OK**.
2. Edit the **\Amazon\Ec2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json** file and configure the types of logs you want to send to CloudWatch Logs. For more information, see Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs (p. 245).

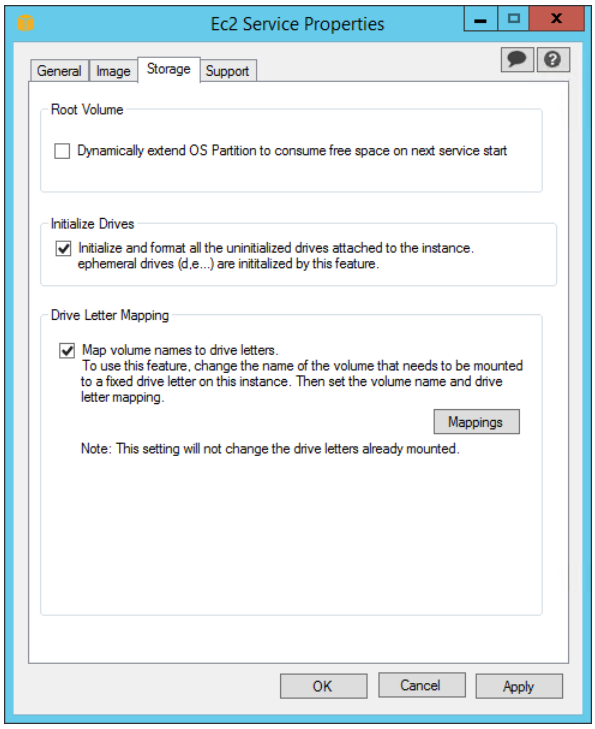
Wallpaper Information

Use this setting to display system information on the desktop background. The following is an example of the information displayed on the desktop background.

```
          Hostname  :  WIN-00J3EXAMPLE
       Instance ID  :  i-2cdbaa52
 Public IP Address  :  203.0.113.17
Private IP Address  :  10.204.22.250
 Availability Zone  :  us-east-1d
     Instance Size  :  m1.large
      Architecture  :  AMD64
      Total Memory  :  7.5 GB
  Processing Power  :  4 ECUs
   I/O Performance  :  High
```

The information displayed on the desktop background is controlled by the settings file `EC2ConfigService\Settings\WallpaperSettings.xml`.

4. Click the **Storage** tab. You can enable or disable the following settings.



Root Volume

This setting dynamically extends Disk 0/Volume 0 to include any unpartitioned space. This can be useful when the instance is booted from a root device volume that has a custom size.

Initialize Drives

This setting formats and mounts all volumes attached to the instance during start.

Drive Letter Mapping

The system maps the volumes attached to an instance to drive letters. For Amazon EBS volumes, the default is to assign drive letters going from D: to Z:. For instance store volumes, the default depends on the driver. Citrix PV drivers assign instance store volumes drive letters going from Z: to A:. Red Hat drivers assign instance store volumes drive letters going from D: to Z:.

To choose the drive letters for your volumes, click **Mappings**. In the **DriveLetterSetting** dialog box, specify the **Volume Name** and **Drive Letter** values for each volume, and then click **OK**. We recommend that you select drive letters that avoid conflicts with drive letters that are likely to be in use, such as drive letters in the middle of the alphabet.

After you specify a drive letter mapping and attach a volume with same label as one of the volume names that you specified, EC2Config automatically assigns your specified drive letter to that volume. However, the drive letter mapping fails if the drive letter is already in use. Note that EC2Config doesn't change the drive letters of volumes that were already mounted when you specified the drive letter mapping.

5. To save your settings and continue working on them later, click **OK** to close the **Ec2 Service Properties** dialog box. If you have finished customizing your instance and want to create an AMI from that instance, see Create a Standard Amazon Machine Image Using Sysprep (p. 90).

EC2Config Settings Files

The settings files control the operation of the EC2Config service. These files are located in the `C:\Program Files\Amazon\Ec2ConfigService\Settings` directory:

- `ActivationSettings.xml`—Controls product activation using a key management server (KMS).
- `AWS.EC2.Windows.CloudWatch.json`—Controls which performance counters to send to CloudWatch and which logs to send to CloudWatch Logs. For more information about how to change the settings in this file, see Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs (p. 245).
- `BundleConfig.xml`—Controls how EC2Config prepares an instance store-backed instance for AMI creation. Note that the only Windows instances that can be backed by instance store are those for Windows Server 2003.
- `Config.xml`—Controls the primary settings.
- `DriveLetterConfig.xml`—Controls drive letter mappings.
- `EventLogConfig.xml`—Controls the event log information that's displayed on the console while the instance is booting.
- `WallpaperSettings.xml`—Controls the information that's displayed on the desktop background.

ActivationSettings.xml

This file contains settings that control product activation. When Windows boots, the EC2Config service checks whether Windows is already activated. If Windows is not already activated, it attempts to activate Windows by searching for the specified KMS server.

- `SetAutodiscover`—Indicates whether to detect a KMS automatically.
- `TargetKMSServer`—Stores the private IP address of a KMS. The KMS must be in the same region as your instance.
- `DiscoverFromZone`—Discovers the KMS server from the specified DNS zone.
- `ReadFromUserData`—Gets the KMS server from UserData.
- `LegacySearchZones`—Discovers the KMS server from the specified DNS zone.
- `DoActivate`—Attempts activation using the specified settings in the section. This value can be `true` or `false`.
- `LogResultToConsole`—Displays the result to the console.

BundleConfig.xml

This file contains settings that control how EC2Config prepares an instance for AMI creation.

- `AutoSysprep`—Indicates whether to use Sysprep automatically. Change the value to `Yes` to use Sysprep.
- `SetRDPCertificate`—Sets a self-signed certificate to the Remote Desktop server running on a Windows 2003 instance. This enables you to securely RDP into the instances. Change the value to `Yes` if the new instances should have the certificate.

  This setting is not used with Windows Server 2008 or Windows Server 2012 instances because they can generate their own certificates.
- `SetPasswordAfterSysprep`—Sets a random password on a newly launched instance, encrypts it with the user launch key, and outputs the encrypted password to the console. Change the value of this setting to `No` if the new instances should not be set to a random encrypted password.

Config.xml

Plug-ins

- `Ec2SetPassword`—Generates a random encrypted password each time you launch an instance. This feature is disabled by default after the first launch so that reboots of this instance don't change a password set by the user. Change this setting to `Enabled` to continue to generate passwords each time you launch an instance.

  This setting is important if you are planning to create an AMI from your instance.
- `Ec2SetComputerName`—Sets the host name of the instance to a unique name based on the IP address of the instance and reboots the instance. To set your own host name, or prevent your existing host name from being modified, you must disable this setting.
- `Ec2InitializeDrives`—Initializes and formats all volumes during startup. This feature is enabled by default.
- `Ec2EventLog`—Displays event log entries in the console. By default, the three most recent error entries from the system event log are displayed. To specify the event log entries to display, edit the `EventLogConfig.xml` file located in the `EC2ConfigService\Settings` directory. For information about the settings in this file, see Eventlog Key in the MSDN Library.
- `Ec2ConfigureRDP`—Sets up a self-signed certificate on the instance, so users can securely access the instance using Remote Desktop. This feature is disabled on Windows Server 2008 and Windows Server 2012 instances because they can generate their own certificates.
- `Ec2OutputRDPCert`—Displays the Remote Desktop certificate information to the console so that the user can verify it against the thumbprint.
- `Ec2SetDriveLetter`—Sets the drive letters of the mounted volumes based on user-defined settings. By default, when an Amazon EBS volume is attached to an instance, it can be mounted using the drive letter on the instance. To specify your drive letter mappings, edit the `DriveLetterConfig.xml` file located in the `EC2ConfigService\Settings` directory.
- `Ec2WindowsActivate`—The plug-in handles Windows activation. It checks to see if Windows is activated. If not, it updates the KMS client settings, and then activates Windows.

  To modify the KMS settings, edit the `ActivationSettings.xml` file located in the `EC2ConfigService\Settings` directory.
- `Ec2DynamicBootVolumeSize`—Extends Disk 0/Volume 0 to include any unpartitioned space.
- `Ec2HandleUserData`—Creates and executes scripts created by the user on the first launch of an instance after Sysprep is run. Commands wrapped in script tags are saved to a batch file, and commands wrapped in PowerShell tags are saved to a .ps1 file.

Global Settings

- `ManageShutdown`—Ensures that instances launched from instance store-backed AMIs do not terminate while running Sysprep.
- `SetDnsSuffixList`—Sets the DNS suffix of the network adapter for Amazon EC2. This allows DNS resolution of servers running in Amazon EC2 without providing the fully qualified domain name.
- `WaitForMetaDataAvailable`—Ensures that the EC2Config service will wait for metadata to be accessible and the network available before continuing with the boot. This check ensures that EC2Config can obtain information from metadata for activation and other plug-ins.
- `ShouldAddRoutes`—Adds a custom route to the primary network adapter to enable the following IP addresses when multiple NICs are attached: 169.254.169.250, 169.254.169.251, and 169.254.169.254. These addresses are used by Windows Activation and when you access instance metadata.
- `RemoveCredentialsfromSyspreponStartup`—Removes the administrator password from `Sysprep.xml` the next time the service starts. To ensure that this password persists, edit this setting.

DriveLetterConfig.xml

This file contains settings that control drive letter mappings. By default, a volume can be mapped to any available drive letter. You can mount a volume to a particular drive letter as follows.

```
<?xml version="1.0" standalone="yes"?>
<DriveLetterMapping>
  <Mapping>
    <VolumeName></VolumeName>
    <DriveLetter></DriveLetter>
  </Mapping>
  . . .
  <Mapping>
    <VolumeName></VolumeName>
    <DriveLetter></DriveLetter>
  </Mapping>
</DriveLetterMapping>
```

- `VolumeName`—The volume label. For example, *`My Volume`*. To specify a mapping for an instance storage volume, use the label `Temporary Storage X`, where `X` is a number from 0 to 25.
- `DriveLetter`—The drive letter. For example, *`M:`*. The mapping fails if the drive letter is already in use.

EventLogConfig.xml

This file contains settings that control the event log information that's displayed on the console while the instance is booting. By default, we display the three most recent error entries from the System event log.

- `Category`—The event log key to monitor.
- `ErrorType`—The event type (for example, `Error`, `Warning`, `Information`.)
- `NumEntries`—The number of events stored for this category.
- `LastMessageTime`—To prevent the same message from being pushed repeatedly, the service updates this value every time it pushes a message.
- `AppName`—The event source or application that logged the event.

WallpaperSettings.xml

This file contains settings that control the information that's displayed on the desktop background. The following information is displayed by default.

- `Hostname`—Displays the computer name.
- `Instance ID`—Displays the ID of the instance.
- `Public IP Address`—Displays the public IP address of the instance.
- `Private IP Address`—Displays the private IP address of the instance.
- `Availability Zone`—Displays the Availability Zone in which the instance is running.
- `Instance Size`—Displays the type of instance.
- `Architecture`—Displays the setting of the `PROCESSOR_ARCHITECTURE` environment variable.
- `AddMemory`—Displays the system memory, in GB.
- `AddECU`—Displays the processing power, in ECU.
- `AddIO`—Displays the I/O performance.

You can remove any of the information that's displayed by default by deleting its entry. You can add additional instance metadata to display as follows.

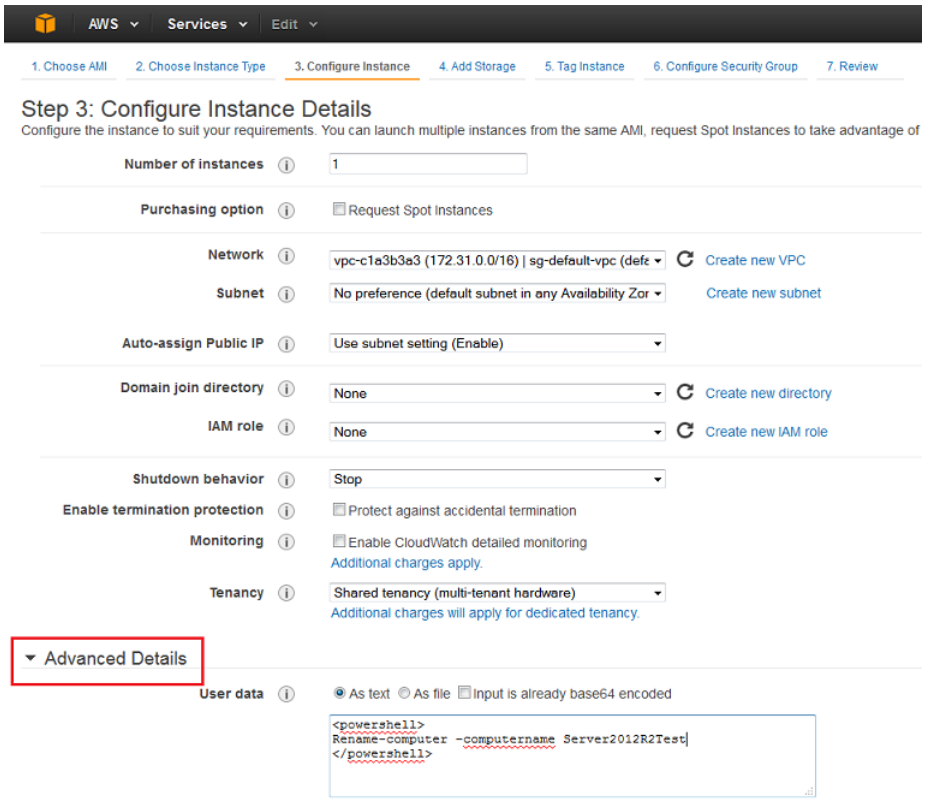
```
<WallpaperInformation>
  <name>display_name</name>
  <source>metadata</source>
  <identifier>meta-data/path</identifier>
</WallpaperInformation>
```

You can add additional System environment variables to display as follows.

```
<WallpaperInformation>
  <name>display_name</name>
  <source>EnvironmentVariable</source>
  <identifier>variable-name</identifier>
</WallpaperInformation>
```

Executing User Data

You can specify scripts to execute when an instance starts. You enter the script in the **User data** section of the Instance Configuration Wizard. The **User data** option is located on the **Configure Instance** page in the **Advanced Details** section. The example in the following image would change the name of the instance to *Server2012R2Test* when the instance booted.



For EC2Config to execute user data, you must enclose the lines of the specified script within one of the following special tags:

`<script></script>`
: Run any command that you can run in a Command Prompt window.

Example: `<script>dir > c:\test.log</script>`

`<powershell></powershell>`
Run any command that you can run at the Windows PowerShell command prompt.

If you use an AMI that includes the AWS Tools for Windows PowerShell, you can also use those cmdlets. If you specify an IAM role when you launch your instance, then you don't need to specify credentials to the cmdlets, as applications that run on the instance can use the role's credentials to access AWS resources such as Amazon S3 buckets.

Example: `<powershell>Read-S3Object -BucketName myS3Bucket -Key myFolder/myFile.zip -File c:\destinationFile.zip</powershell>`

You can separate the commands in a script using line breaks.

If EC2Config finds `script` or `powershell` tags, it saves the script to a batch or PowerShell file in its `/Scripts` folder. It runs these files when the instance starts. If both `script` and `powershell` tags are present, it runs the batch script first and the PowerShell script next, regardless of the order in which they appear.

The `/Logs` folder contains output from the standard output and standard error streams.

EC2Config expects the user data to be available in base64 encoding. If the user data is not available in base64 encoding, EC2Config logs an error about being unable to find `script` or `powershell` tags to execute. If your encoding is not correct, the following is an example that sets the encoding using PowerShell.

```
$UserData = [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.Get
Bytes($Script))
```

Initial Boot

By default, all Amazon AMIs have user data execution enabled for the initial boot. If you click **Shutdown with Sysprep** in EC2Config, user data execution is enabled, regardless of the setting of the **User Data** check box.

User data execution happens under the local administrator user only when a random password is generated. This is because EC2Config generates the password and is aware of the credentials briefly (prior to sending to the console). EC2Config doesn't store or track password changes, so when you don't generate a random password, user data execution is performed by the EC2Config service account.

Subsequent Boots

Because Amazon AMIs automatically disable user data execution after the initial boot, you must do one of the following to make user data persist across reboots:

- Programmatically create a scheduled task to run at system start using `schtasks.exe /Create`, and point the scheduled task to the user data script (or another script) at `C:\Program Files\Amazon\Ec2ConfigServer\Scripts\UserScript.ps1`.
- Programmatically enable the user data plug-in in `Config.xml` using a script similar to the following:

```
<powershell>
$EC2SettingsFile="C:\Program Files\Amazon\Ec2ConfigService\Settings\Config.xml"
$xml = [xml](get-content $EC2SettingsFile)
$xmlElement = $xml.get_DocumentElement()
$xmlElementToModify = $xmlElement.Plugins
```

```
foreach ($element in $xmlElementToModify.Plugin)
{
    if ($element.name -eq "Ec2SetPassword")
    {
        $element.State="Enabled"
    }
    elseif ($element.name -eq "Ec2HandleUserData")
    {
        $element.State="Enabled"
    }
}
$xml.Save($EC2SettingsFile)
</powershell>
```

- Starting with EC2Config version 2.1.10, you can use `<persist>true</persist>` to enable the plug-in after user data execution.

```
<powershell>
    insert script here
</powershell>
<persist>true</persist>
```

Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs

Starting with EC2Config version 2.2.5 (version 2.2.6 or later is recommended), you can export all Windows Server messages in the system, security, application, and IIS logs to CloudWatch Logs and monitor them using CloudWatch metrics. EC2Config version 2.2.10 or later adds the ability to export any event log data, Event Tracing (Windows), or text-based log files to CloudWatch Logs. In addition, you can also export performance counter data to CloudWatch. To manage the performance counters and logs for multiple instances, you can use Amazon EC2 Simple Systems Manager (SSM). For more information, see Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs (p. 299).

To set up EC2Config to send data to CloudWatch Logs, complete the following steps:

Topics

- Step 1: Configure IAM Permissions (p. 245)
- Step 2: Enable CloudWatch Logs Integration (p. 246)
- Step 3: Configure the Credentials for CloudWatch and CloudWatch Logs (p. 248)
- Step 4: Configure the Performance Counters and Logs to Send to CloudWatch and CloudWatch Logs (p. 249)
- Step 5: Configure the Flow Control (p. 255)
- Step 6: Restart EC2Config (p. 255)
- Troubleshooting CloudWatch Logs in EC2Config (p. 255)

Step 1: Configure IAM Permissions

You can use the following IAM permissions in an instance profile attached to an Amazon EC2 instance when you launch the instance. EC2Config uses the instance profile when uploading CloudWatch metrics or logs to CloudWatch Logs. For more information about instance profiles, see Instance Profiles in the

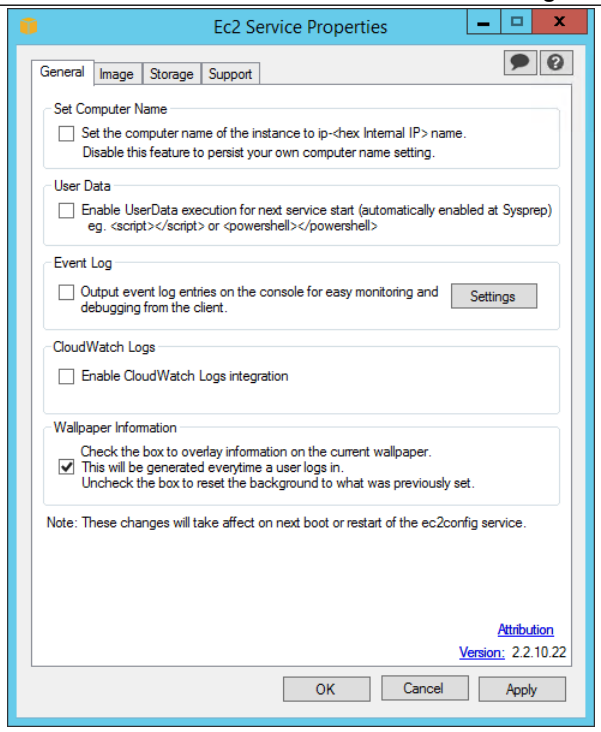
IAM User Guide. For more information about launching an instance with an IAM role, see IAM Roles for Amazon EC2 (p. 442).

Note
These IAM permissions only work with the local JSON configuration file. If you want to upload logs through Amazon EC2 Simple Systems Manager (SSM), see Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs (p. 299).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessToSSM",
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Step 2: Enable CloudWatch Logs Integration

1. Launch and connect to your Windows instance.
2. From the **Start** menu, click **All Programs**, and then click **EC2ConfigService Settings**.



3. On the **General** tab of the **Ec2 Service Properties** dialog box, under **CloudWatch Logs**, select **Enable CloudWatch Logs integration**, and then click **OK**.
4. Create a configuration file named **AWS.EC2.Windows.CloudWatch.json**.

   To download a sample of the file, see AWS.EC2.Windows.CloudWatch.json.

Note

You can also enable CloudWatch Logs by adding the following script to the user data field when you launch an instance. EC2Config will run this script every time your instance is restarted to make sure that CloudWatch Logs integration is enabled. To run this script only when an instance is first launched, remove <persist>true</persist> from the script.

```
<powershell>
$EC2SettingsFile="C:\Program Files\Amazon\Ec2ConfigService\Settings\Con
fig.xml"
$xml = [xml](get-content $EC2SettingsFile)
$xmlElement = $xml.get_DocumentElement()
$xmlElementToModify = $xmlElement.Plugins

foreach ($element in $xmlElementToModify.Plugin)
{
    if ($element.name -eq "AWS.EC2.Windows.CloudWatch.PlugIn")
    {
        $element.State="Enabled"
    }
}
$xml.Save($EC2SettingsFile)
</powershell>
<persist>true</persist>
```

Step 3: Configure the Credentials for CloudWatch and CloudWatch Logs

To set the credentials, region, and metric namespace for CloudWatch

This section of the JSON file defines the credentials, region, and metric namespace that comprise the destination where your data is sent. You can add additional sections with unique IDs (for example, "CloudWatch2", CloudWatch3", etc.) and specify a different region for each new ID to send the same data to different locations.

Note
You only need to set CloudWatch credentials if you are using EC2Config and plan to send performance counters to CloudWatch. If you're using Amazon EC2 Simple Systems Manager, your credentials are configured in the IAM role you used when you launched your Amazon EC2 instance.

1. In the JSON file, locate the **CloudWatch** section.

```
{
    "Id": "CloudWatch",
    "FullName": "AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputCom
ponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "AccessKey": "",
        "SecretKey": "",
        "Region": "us-west-1",
        "NameSpace": "Windows/Default"
    }
},
```

2. In the **AccessKey** parameter, enter your access key ID. This is not supported if you launched your instance using an IAM role. For more information, see IAM Roles for Amazon EC2 (p. 442).
3. In the **SecretKey** parameter, enter your secret access key. This is not supported if you launched your instance using an IAM role. For more information, see IAM Roles for Amazon EC2 (p. 442).
4. In the **Region** parameter, enter the region where you want to send log data. You can specify us-east-1, us-west-1, us-west-2, eu-west-1, eu-central-1, ap-southeast-1, ap-southeast-2, or ap-northeast-1. Although you can send performance counters to a different region from where you send your log data, we recommend that you set this parameter to the same region where your instance is running.
5. In the **NameSpace** parameter, enter the metric namespace where you want performance counter data to be written in CloudWatch.

To set the credentials, region, log group, and log stream for CloudWatch Logs

This section of the JSON file defines the credentials, region, log group name and log stream namespace that comprise the destination where your data is sent. You can add additional sections with unique IDs (for example, "CloudWatchLogs2", CloudWatchLogs3", etc.) and specify a different region for each new ID to send the same data to different locations.

1. In the JSON file, locate the **CloudWatchLogs** section.

```
{
    "Id": "CloudWatchLogs",
    "FullName": "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Win
dows.CloudWatch",
```

```
    "Parameters": {
        "AccessKey": "",
        "SecretKey": "",
        "Region": "us-east-1",
        "LogGroup": "Default-Log-Group",
        "LogStream": "{instance_id}"
    }
},
```

2. In the **AccessKey** parameter, enter your access key ID. This is not supported if you launched your instance using an IAM role. For more information, see IAM Roles for Amazon EC2 (p. 442).
3. In the **SecretKey** parameter, enter your secret access key. This is not supported if you launched your instance using an IAM role. For more information, see IAM Roles for Amazon EC2 (p. 442).
4. In the **Region** parameter, enter the region where you want EC2Config to send log data. You can specify us-east-1, us-west-1, us-west-2, eu-west-1, eu-central-1, ap-southeast-1, ap-southeast-2, or ap-northeast-1.
5. In the **LogGroup** parameter, enter the name for your log group. This is the same name that will be displayed on the **Log Groups** screen in the CloudWatch console.
6. In the **LogStream** parameter, enter the destination log stream. If you use **{instance_id}**, the default, EC2Config uses the instance ID of this instance as the log stream name.

   If you enter a log stream name that doesn't already exist, CloudWatch Logs automatically creates it for you. You can use a literal string or predefined variables (**{instance_id}**, **{hostname}**, **{ip_address}**, or a combination of all three to define a log stream name.

   The log stream name specified in this parameter appears on the **Log Groups > Streams for *<YourLogStream>*** screen in the CloudWatch console.

Step 4: Configure the Performance Counters and Logs to Send to CloudWatch and CloudWatch Logs

To configure the performance counters to send to CloudWatch

You can select any performance counters that are available in Performance Monitor. You can select different categories to upload to CloudWatch as metrics, such as .NET CLR Data, ASP.NET Applications, HTTP Service, Memory, or Process and Processors.

For each performance counter that you want to upload to CloudWatch, copy the **PerformanceCounter** section and change the **Id** parameter to make it unique (e.g., "PerformanceCounter2") and update the other parameters as necessary.

1. In the JSON file, locate the **PerformanceCounter** section.

```
{
    "Id": "PerformanceCounter",
   "FullName": "AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.Per
formanceCounterInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "CategoryName": "Memory",
        "CounterName": "Available MBytes",
        "InstanceName": "",
        "MetricName": "AvailableMemory",
        "Unit": "Megabytes",
        "DimensionName": "",
```

```
        "DimensionValue": ""
    }
},
```

2. In the **CategoryName** parameter, enter the performance counter category.

   a. To find the available categories and counters, open Performance Monitor.
   b. Click **Monitoring Tools**, and then click **Performance Monitor**.
   c. In the results pane, click the green **+** (plus) button.

      The categories and counters are listed in the **Add Counters** dialog box.

3. In the **CounterName** parameter, enter the name of the performance counter.
4. In the **InstanceName** parameter, in the **Add Counters** dialog box in Performance Monitor, enter one of the **Instances of selected object**. Do not use an asterisk (*) to indicate all instances because each performance counter component only supports one metric. You can, however use **_Total**.
5. In the **MetricName** parameter, enter the CloudWatch metric that you want performance data to appear under.
6. In the **Unit** parameter, enter the appropriate unit of measure for the metric:

   Seconds | Microseconds | Milliseconds | Bytes | Kilobytes | Megabytes | Gigabytes | Terabytes | Bits | Kilobits | Megabits | Gigabits | Terabits | Percent | Count | Bytes/Second | Kilobytes/Second | Megabytes/Second | Gigabytes/Second | Terabytes/Second | Bits/Second | Kilobits/Second | Megabits/Second | Gigabits/Second | Terabits/Second | Count/Second | None.
7. (optional) You can enter a dimension name and value in the **DimensionName** and **DimensionValue** parameters to specify a dimension for your metric. These parameters provide another view when listing metrics. You can also use the same dimension for multiple metrics so that you can view all metrics belonging to a specific dimension.

To send Windows application event log data to CloudWatch Logs

1. In the JSON file, locate the **ApplicationEventLog** section.

```
{
    "Id": "ApplicationEventLog",
    "FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputCompon
ent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Application",
        "Levels": "1"
    }
},
```

2. In the **Levels** parameter, enter one of the following values:

   1 - Only error messages uploaded.

   2 - Only warning messages uploaded.

   4 - Only information messages uploaded.

   You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send security log data to CloudWatch Logs

1. In the JSON file, locate the **SecurityEventLog** section.

```
{
    "Id": "SecurityEventLog",
    "FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputCompon
ent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Security",
        "Levels": "7"
    }
},
```

2. In the **Levels** parameter, enter one of the following values:

   1 - Only error messages uploaded.

   2 - Only warning messages uploaded.

   4 - Only information messages uploaded.

   You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send system event log data to CloudWatch Logs

1. In the JSON file, locate the **SystemEventLog** section.

```
{
    "Id": "SystemEventLog",
    "FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputCompon
ent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
        "Levels": "7"
    }
},
```

2. In the **Levels** parameter, enter one of the following values:

   1 - Only error messages uploaded.

   2 - Only warning messages uploaded.

   4 - Only information messages uploaded.

   You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send other types of event log data to CloudWatch Logs

In addition to the application, system, and security logs, you can upload other types of event logs.

1. In the JSON file, add a new section.

```
{
    "Id": "",
    "FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputCompon
ent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "",
        "Levels": "7"
    }
},
```

2. In the **Id** parameter, enter a name for the log you want to upload (e.g., WindowsBackup).
3. In the **LogName** parameter, enter the name of the log you want to upload.

   a. To find the name of the log, in Event Viewer, in the navigation pane, click **Applications and Services Logs**.
   b. In the list of logs, right-click the log you want to upload (e.g., Microsoft>Windows>Backup>Operational), and then click **Create Custom View**.
   c. In the **Create Custom View** dialog box, click the **XML** tab. The **LogName** is in the <Select Path=> tag (e.g., `Microsoft-Windows-Backup`). Copy this text into the **LogName** parameter in the **AWS.EC2.Windows.CloudWatch.json** file.

4. In the **Levels** parameter, enter one of the following values:

   1 - Only error messages uploaded.

   2 - Only warning messages uploaded.

   4 - Only information messages uploaded.

   You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send Event Tracing (Windows) data to CloudWatch Logs

ETW (Event Tracing for Windows) provides an efficient and detailed logging mechanism that applications can write logs to. Each ETW is controlled by a session manager that can start and stop the logging session. Each session has a provider and one or more consumers.

1. In the JSON file, locate the **ETW** section.

```
{
    "Id": "ETW",
    "FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputCompon
ent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Microsoft-Windows-WinINet/Analytic",
        "Levels": "7"
    }
},
```

2. In the **LogName** parameter, enter the name of the log you want to upload.

a. To find the name of the log, in Event Viewer, on the **View** menu, click **Show Analytic and Debug Logs**.
b. In the navigation pane, click **Applications and Services Logs**.
c. In the list of ETW logs, right-click the log you want to upload, and then click **Enable Log**.
d. Right-click the log again, and click **Create Custom View**.
e. In the **Create Custom View** dialog box, click the **XML** tab. The **LogName** is in the <Select Path=> tag (e.g., `Microsoft-Windows-WinINet/Analytic`). Copy this text into the **LogName** parameter in the **AWS.EC2.Windows.CloudWatch.json** file.

3. In the **Levels** parameter, enter one of the following values:

   1 - Only error messages uploaded.

   2 - Only warning messages uploaded.

   4 - Only information messages uploaded.

   You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send custom logs (any text-based log file) to CloudWatch Logs

1. In the JSON file, locate the **CustomLogs** section.

```
{
    "Id": "CustomLogs",
    "FullName": "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputCompon
ent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\CustomLogs\\",
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "Local",
        "LineCount": "5"
    }
},
```

2. In the **LogDirectoryPath** parameter, enter the path where logs are stored on your instance.
3. In the **TimestampFormat** parameter, enter the timestamp format you want to use. For a list of supported values, see the Custom Date and Time Format Strings topic on MSDN.

   Note
   Your source log file must have the timestamp at the beginning of each log line.

4. In the **Encoding** parameter, enter the file encoding to use (e.g., UTF-8). For a list of supported values, see the Encoding Class topic on MSDN.

   Note
   Use the encoding name, not the display name, as the value for this parameter.

5. (optional) In the **Filter** parameter, enter the prefix of log names. Leave this parameter blank to monitor all files. For a list of supported values, see the FileSystemWatcherFilter Property topic on MSDN.

6. (optional) In the **CultureName** parameter, enter the locale where the timestamp is logged. If **CultureName** is blank, it defaults to the same locale currently used by your Windows instance. For a list of supported values, see the National Language Support (NLS) API Reference topic on MSDN.

   Note
   The **div**, **div-MV**, **hu**, and **hu-HU** values are not supported.

7. (optional) In the **TimeZoneKind** parameter, enter **Local** or **UTC**. You can set this to provide time zone information when no time zone information is included in your log's timestamp. If this parameter is left blank and if your timestamp doesn't include time zone information, CloudWatch Logs defaults to the local time zone. This parameter is ignored if your timestamp already contains time zone information.
8. (optional) In the **LineCount** parameter, enter the number of lines in the header to identify the log file. For example, IIS log files have virtually identical headers. You could enter **3**, which would read the first three lines of the log file's header to identify it. In IIS log files, the third line is the date and time stamp, which is different between log files.

To send IIS log data to CloudWatch Logs

1. In the JSON file, locate the **IISLog** section.

```
{
    "Id": "IISLogs",
    "FullName": "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputCompon
ent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "UTC",
        "LineCount": "3"
    }
},
```

2. In the **LogDirectoryPath** parameter, enter the folder where IIS logs are stored for an individual site (e.g., **C:\\inetpub\\logs\\LogFiles\\W3SVC*n***).

   Note
   Only W3C log format is supported. IIS, NCSA, and Custom formats are not supported.

3. In the **TimestampFormat** parameter, enter the timestamp format you want to use. For a list of supported values, see the Custom Date and Time Format Strings topic on MSDN.
4. In the **Encoding** parameter, enter the file encoding to use (e.g., UTF-8). For a list of supported values, see the Encoding Class topic on MSDN.

   Note
   Use the encoding name, not the display name, as the value for this parameter.

5. (optional) In the **Filter** parameter, enter the prefix of log names. Leave this parameter blank to monitor all files. For a list of supported values, see the FileSystemWatcherFilter Property topic on MSDN.
6. (optional) In the **CultureName** parameter, enter the locale where the timestamp is logged. If **CultureName** is blank, it defaults to the same locale currently used by your Windows instance. For a list of supported values, see the National Language Support (NLS) API Reference topic on MSDN.

   Note
   The **div**, **div-MV**, **hu**, and **hu-HU** values are not supported.

7. (optional) In the **TimeZoneKind** parameter, enter **Local** or **UTC**. You can set this to provide time zone information when no time zone information is included in your log's timestamp. If this parameter is left blank and if your timestamp doesn't include time zone information, CloudWatch Logs defaults to the local time zone. This parameter is ignored if your timestamp already contains time zone information.
8. (optional) In the **LineCount** parameter, enter the number of lines in the header to identify the log file. For example, IIS log files have virtually identical headers. You could enter **3**, which would read the first three lines of the log file's header to identify it. In IIS log files, the third line is the date and time stamp, which is different between log files.

Step 5: Configure the Flow Control

In order to send performance counter data to CloudWatch or to send log data to CloudWatch Logs, each data type must have a corresponding destination listed in the **Flows** section. For example, to send a performance counter defined in the **"Id": "PerformanceCounter"** section of the JSON file to the CloudWatch destination defined in the **"Id": "CloudWatch"** section of the JSON file, you would enter **"PerformanceCounter,CloudWatch"** in the **Flows** section. Similarly, to send the custom log, ETW log, and system log to CloudWatch Logs, you would enter **"(CustomLogs, ETW,SystemEventLog),CloudWatchLogs"**. In addition, you can send the same performance counter or log file to more than one destination. For example, to send the application log to two different destinations that you defined in the **"Id": "CloudWatchLogs"** section of the JSON file, you would enter **"ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"** in the **Flows** section.

1. In the JSON file, locate the **Flows** section.

```
"Flows": {
    "Flows": [
      "PerformanceCounter,CloudWatch",
      "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
      "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
      "CustomLogs, CloudWatchLogs2",
      "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
    ]
}
```

2. In the **Flows** parameter, enter each data type that you want to upload (e.g., ApplicationEventLog) and destination where you want to send it (e.g., CloudWatchLogs).

Step 6: Restart EC2Config

After you're finished updating the **C:\Program Files\Amazon\Ec2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json** file, you should restart EC2Config. For more information, see Stopping, Restarting, Deleting, or Uninstalling EC2Config (p. 258).

Troubleshooting CloudWatch Logs in EC2Config

If you're experiencing trouble with uploading performance counters or logs, the first place you should check is the **C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2ConfigLog.txt** file. Some of the most commonly encountered problems are listed below.

I cannot see logs in the CloudWatch console.

Please verify that you are using EC2Config version 2.2.6 or later. If you are still using EC2Config version 2.2.5, use the following steps to solve the issue:

1. In the Services Microsoft Management Console (MMC) snap-in, restart the EC2Config service. To open the **Services** snap-in, click the **Start** menu and then in the **Run** box, type **services.msc**.
2. Sign in to the AWS Management Console and open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
3. On the navigation bar, select the appropriate region.
4. In the navigation pane, click **Logs**.
5. In the contents pane, in the **Expire Events After** column, click the retention setting for the log group that you just created.
6. In the **Edit Retention** dialog box, in the **New Retention** list, select **10 years (3653 days)**, and then click **OK**.

   Note
   You can also set log retention (in days) using the following Windows PowerShell command:

```
Write-CWLRetentionPolicy-LogGroupName Default-Log-Group -Reten
tionInDays 3653
```

The Enable CloudWatch Logs integration check box won't stay selected after I click OK and then reopen EC2Config.
This issue might occur if you've performed an upgrade from an earlier version of EC2Config to version 2.2.5. To resolve this issue, install version 2.2.6 or later.

I see errors like *Log events cannot be more than 2 hours in the future* or *InvalidParameterException*.
This error might occur if you are using EC2Config version 2.2.5 and your instance's time zone falls between UTC-12:00 and UTC-02:00. To resolve this issue, install EC2Config version 2.2.6 or later.

I cannot see SQL Server logs in the CloudWatch console and see this error in Ec2ConfigLog.txt *[Error] Exception occurred: Index and length must refer to a location within the string. Parameter name: length*.
To resolve this issue, install EC2Config version 2.2.11 or later.

I'm running ten or fewer workflows and EC2Config is using over 500MB of memory.
To resolve this issue, install version 2.3.313 or later.

Only the first one or two IIS logs are uploaded and then no other IIS logs get uploaded.
Update the **IISlog** section of the **C:\Program Files\Amazon\Ec2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json** file and set the **LineCount** parameter to **3**, which would read the first three lines of the log file's header to identify it. In IIS log files, the third line is the date and time stamp, which is different between log files.

Installing the Latest Version of EC2Config

By default, the EC2Config service is included in each AWS Windows AMI. When we release an updated version, we update all AWS Windows AMIs with the latest version. However, you need to update your own Windows AMIs and instances with the latest version.

To find notifications of updates to EC2Config, go to the Amazon EC2 forum. For more information about the changes in each version, see the What's New section on the download page.

To verify the version of EC2Config included with your Windows AMI

1. Launch an instance from your AMI and connect to it.
2. In Control Panel, select **Programs and Features**.
3. In the list of installed programs, look for `Ec2ConfigService`. Its version number appears in the **Version** column.

To install the latest version of EC2Config on your instance

1. (Optional) If you have changed any settings, note these changes, as you'll need to restore them after installing the latest version of EC2Config.
2. Go to Amazon Windows EC2Config Service.
3. Click **Download**.
4. Download and unzip the file.
5. Run `EC2Install.exe`. For a complete list of options, run `EC2Install` with the `/?` option. Note the following:

   - By default, the setup replaces your settings files with default settings files during installation and restarts the EC2Config service when the installation is completed. To keep the custom settings that you saved in step 1, run `EC2Install` with the `/norestart` option, restore your settings, and then restart the EC2Config service manually.
   - By default, the setup displays prompts. To run the command with no prompts, use the `/quiet` option.

6. Connect to your instance, run the Services administrative tool, and verify that the status of `EC2Config service` is `Started`.

If you can't connect to your instance, it's possible that updating its version of EC2Config will solve the issue. If your instance is an Amazon EBS-backed instance, you can use the following procedure to update EC2Config even though you can't connect to your instance.

To update EC2Config on an Amazon EBS-backed Windows instance that you can't connect to

1. Stop the affected instance and detach its root volume.
2. Launch a temporary `t2.micro` instance in the same Availability Zone as the affected instance using an AMI for Windows Server 2003. (If you use a later version of Windows Server, you won't be able to boot the original instance when you restore its root volume.) To find an AMI for Windows Server 2003, search for public Windows AMIs with the name `Windows_Server-2003-R2_SP2`.
3. Attach the root volume from the affected instance to this temporary instance. Connect to the temporary instance, open the **Disk Management** utility, and bring the drive online.
4. Download the latest EC2Config from Amazon Windows EC2Config Service. Extract the files from the `.zip` file to the `Temp` directory on the drive you attached.
5. Open **Regedit** and select **HKEY_LOCAL_MACHINE**. From the **File** menu, click **Load Hive**. Select the drive, open the file `Windows\System32\config\SOFTWARE`, and specify a key name when prompted (you can use any name).
6. Select the key you just loaded and navigate to `Microsoft\Windows\CurrentVersion`. Select the `RunOnce` key. (If this key doesn't exist, right-click `CurrentVersion`, point to **New**, select **Key**, and name the key `RunOnce`.) Right-click, point to **New**, and select **String Value**. Enter `Ec2Install` as the name and `C:\Temp\Ec2Install.exe /quiet` as the data.
7. Select the key again, and from the **File** menu, click **Unload Hive**.
8. Open the **Disk Management** utility and bring the drive offline. Detach the volume from the temporary instance. You can terminate the temporary instance if you have no further use for it.
9. Restore the root volume of the affected instance by attaching it as `/dev/sda1`.
10. Start the instance.
11. After the instance starts, check the system log and verify that you see the message `Windows is ready to use`.

Stopping, Restarting, Deleting, or Uninstalling EC2Config

You can manage the EC2Config service just as you would any other service.

To apply updated settings to your instance, you can stop and restart the service. If you're manually installing EC2Config, you must stop the service first.

To stop the EC2Config service

1. Launch and connect to your Windows instance.
2. On the **Start** menu, point to **Administrative Tools**, and then click **Services**.
3. In the list of services, right-click **EC2Config**, and select **Stop**.

To restart the EC2Config service

1. Launch and connect to your Windows instance.
2. On the **Start** menu, point to **Administrative Tools**, and then click **Services**.
3. In the list of services, right-click **EC2Config**, and select **Restart**.

If you don't need to update the configuration settings, create your own AMI, or use Amazon EC2 Simple Systems Manager (SSM), you can delete and uninstall the service. Deleting a service removes its registry subkey. Uninstalling a service removes the files, the registry subkey, and any shortcuts to the service.

To delete the EC2Config service

1. Start a command prompt window.
2. Run the following command:

```
C:\> sc delete ec2config
```

To uninstall EC2Config

1. Launch and connect to your Windows instance.
2. On the **Start** menu, click **Control Panel**.
3. Double-click **Programs and Features**.
4. On the list of programs, select **EC2ConfigService**, and click **Uninstall** .

Paravirtual Drivers

Amazon Windows AMIs contain a set of drivers to permit access to Xen virtualized hardware. These drivers are used by Amazon EC2 to map instance store and Amazon EBS volumes to their devices. The following table shows key differences between the different drivers.

	RedHat PV	SUSE PV	Windows PV
Instance type	Not supported for all instance types. If you specify an unsupported instance type, the instance is impaired.	Supported for all instance types.	Supported for all instance types.
Attached volumes	Supports up to 16 attached volumes.	Supports up to 16 attached volumes.	Supports up to 16 attached volumes.

	RedHat PV	SWX iV	MP P
Network	The driver has known issues where the network connection resets under high loads; for example, fast FTP file transfers.		

	RedHat PV	SUSE PV	Windows PV

	RedHat PV	SWMS xiP VP
		.pdg

Contents

- AWS PV Drivers (p. 262)
- Citrix PV Drivers (p. 264)
- RedHat PV Drivers (p. 264)

Drivers According to Windows Version

The following list shows which PV drivers are running on each version of Windows Server in AWS.

- Windows Server 2003, Citrix 5.9
- Windows Server 2003 R2, Citrix 5.9
- Windows Server 2008, Citrix 5.9
- Windows Server 2008 R2, Citrix 5.9
- Windows Server 2012 RTM, Citrix 5.9
- Windows Server 2012 R2, AWS PV

AWS PV Drivers

Windows Server 2012 R2 AMIs include AWS PV drivers. The AWS PV drivers are stored in the `%ProgramFiles%\Amazon\Xentools` directory. This directory also contains public symbols and a command line tool, `xenstore_client.exe`, that enables you to access entries in XenStore. For example, the following PowerShell command returns the current time from the Hypervisor:

```
[DateTime]::FromFileTimeUTC((gwmi -n root\wmi -cl AWSXenStoreBase).XenTime).To
String("hh:mm:ss")
11:17:00
```

The AWS PV driver components are listed in the Windows registry under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services`. These driver components are as follows: XENBUS, xeniface, xennet, xenvbd, and xenvif.

AWS PV also has a driver component named LiteAgent, which runs as a Windows service. It handles tasks such as shutdown and restart events from the API. You can access and manage services by running `Services.msc` from the command line.

AWS PV Drivers Version History

The following table describes the version and corresponding changes to AWS PV drivers.

Version	Description
7.3.2	**Logging and diagnostics:** Improved logging and diagnostics. **Stability fix in AWS PV Storage driver:** In some cases the disks may not surface in Windows after reattaching the disk to the instance.
7.3.1	**TRIM update:** Fix related to TRIM requests. This fix stabilizes instances and improves instance performance when managing large numbers of TRIM requests.
7.3.0	**TRIM support:** The AWS PV driver now sends TRIM requests to the hypervisor. Ephemeral disks will properly process TRIM requests given the underlying storage supports TRIM (SSD). Note that EBS-based storage does not support TRIM as of March 2015.
7.2.5	**Stability fix in AWS PV Storage drivers:** In some cases the AWS PV driver could dereference invalid memory and cause a system failure. **Stability fix while generating a crash dump:** In some cases the AWS PV driver could get stuck in a race condition when writing a crash dump. Before this release, the issue could only be resolved by forcing the driver to stop and restart which lost the memory dump.
7.2.4	**Device ID persistence:** This driver fix masks the platform PCI device ID and forces the system to always surface the same device ID, even if the instance is moved. More generally, the fix affects how the hypervisor surfaces virtual devices. The fix also includes modifications to the co-installer for the AWS PV drivers so the system persists mapped virtual devices.
7.2.2	**Load the AWS PV drivers in Directory Services Restore Mode (DSRM) mode:** Directory Services Restore Mode is a safe mode boot option for Windows Server domain controllers. **Persist device ID when virtual network adapter device is reattached:** This fix forces the system to check the MAC address mapping and persist the device ID. This fix ensures that adapters retain their static settings if the adapters are reattached.

Version	Description
7.2.1	**Run in safe mode:** Fixed an issue where the driver would not load in safe mode. Previously the AWS PV Drivers would only instantiate in normal running systems. **Add disks to Microsoft Windows Storage Pools:** Previously we synthesized page 83 queries. The fix disabled page 83 support. Note this does not affect storage pools that are used in a cluster environment because PV disks are not valid cluster disks.
7.2.0	**Base:** The AWS PV base version.

Citrix PV Drivers

The Citrix drivers are stored in the `%ProgramFiles%\Citrix\XenTools` (32-bit instances) or `%ProgramFiles(x86)%\Citrix\XenTools` (64-bit instances) directory.

The Citrix driver components are listed in the Windows registry under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services`. These driver components are as follows: xenevtchn, xeniface, xennet, Xennet6, xensvc, xenvbd, and xenvif.

Citrix also has a driver component named XenGuestAgent, which runs as a Windows service. It handles tasks such as time synchronization at boot (Windows Server 2003 only), and shutdown and restart events from the API. You can access and manage services by running `Services.msc` from the command line.

If you are encountering networking errors while performing certain workloads, you may need to disable the TCP offloading feature for the Citrix PV driver. For more information, see TCP Offloading (p. 274).

RedHat PV Drivers

RedHat drivers are supported for legacy instances, but are not recommended on newer instances with more than 12GB of RAM due to driver limitations. Instances with more than 12GB of RAM running RedHat drivers can fail to boot and become inaccessible. We recommend upgrading RedHat drivers to Citrix PV or AWS PV drivers.

The source files for the RedHat drivers are in the `%ProgramFiles%\RedHat` (32-bit instances) or `%ProgramFiles(x86)%\RedHat` (64-bit instances) directory. The two drivers are `rhelnet`, the RedHat Paravirtualized network driver, and `rhelscsi`, the RedHat SCSI miniport driver.

Related Topics

Upgrade: For more information about upgrading PV drivers, see Upgrading PV Drivers on Your Windows AMI (p. 265).

Troubleshooting: For more information about troubleshooting EC2 drivers, see Troubleshooting PV Drivers (p. 270). For information about troubleshooting EC2 Windows instances, see Troubleshooting Windows Instances (p. 687).

Upgrading PV Drivers on Your Windows AMI

If your Windows instance is launched from a Windows Server 2012 R2 AMI, it uses AWS PV drivers. If your Windows instance uses RedHat drivers, you can upgrade to Citrix drivers. If you are already using Citrix drivers, you can upgrade the Citrix Xen guest agent service. To verify which driver your Windows instance uses, open **Network Connections** in Control Panel and view the **Local Area Connection**. Check whether the driver is one of the following:

- `AWS PV Network Device`
- `Citrix PV Ethernet Adapter`
- `RedHat PV NIC Driver`

Alternatively, you can check the output from the `pnputil -e` command.

Contents

- Upgrade AWS PV Drivers (p. 265)
- Upgrading PV Drivers on Your Windows Server 2008 and 2008 R2 Instances (p. 266) (RedHat to Citrix upgrade)
- Upgrading Your Citrix Xen Guest Agent Service (p. 267)
- Upgrading PV Drivers on Your Windows Server 2003 Instance (p. 268) (RedHat to Citrix upgrade)

Upgrade AWS PV Drivers

Use the following procedure to perform an in-place upgrade of AWS PV Drivers. This procedure does not affect Citrix or RedHat drivers. See the other procedures on this page to upgrade Citrix or RedHat drivers.

To perform an in-place driver upgrade

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Choose the instance that requires the in-place driver upgrade, open the context (right-click) menu, choose **Instance State**, and then choose **Stop**.

   Warning
   When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.
4. After the instance is stopped create a backup. Open the context (right-click) menu for the instance, choose **Image**, and then choose **Create Image**.
5. From the context (right-click) menu for the instance, choose **Instance State**, and then choose **Start**.
6. Connect to the instance using Remote Desktop and then download https://s3.amazonaws.com/ec2-downloads-windows/Drivers/AWSPVDriverSetup.zip to the instance.
7. Extract the contents of the folder and then run `AWSPVDriverSetup.msi`.

After running the MSI, the instance automatically reboots and then upgrades the drivers. The instance will not be available for up to 15 minutes. After the upgrade is complete and the instance passes both health checks in the Amazon EC2 console, connect to the instance using Remote Desktop and verify that the new drivers were installed. In Device Manager, under **Storage Controllers**, locate **AWS PV Storage Host Adapter**. Check the driver properties for version 7.3.2.

Upgrading PV Drivers on Your Windows Server 2008 and 2008 R2 Instances

Before you start upgrading your RedHat drivers to Citrix drivers, make sure you do the following:

- Install the latest version of EC2Config by going to Amazon Windows EC2Config Service. For more information about the EC2Config service, see Configuring a Windows Instance Using the EC2Config Service (p. 235).
- Verify that you have Windows PowerShell 2.0 installed. To verify the version that you have installed, run the following command in a PowerShell window:

```
PS C:> $PSVersionTable.PSVersion
```

  If you need to install version 2.0, see Windows Management Framework (Windows PowerShell 2.0, WinRM 2.0, and BITS 4.0) from Microsoft Support.
- Back up your important information on the instance, or create an AMI from the instance. For more information about creating an AMI, see Creating an Amazon EBS-Backed Windows AMI (p. 68). If you create an AMI, make sure that you do the following:
  - Write down your password.
  - Do not run the Sysprep tool manually or using the EC2Config service.
  - Set your Ethernet adapter to obtain an IP address automatically using DHCP. For more information, see Configure TCP/IP Settings in the Microsoft TechNet Library.

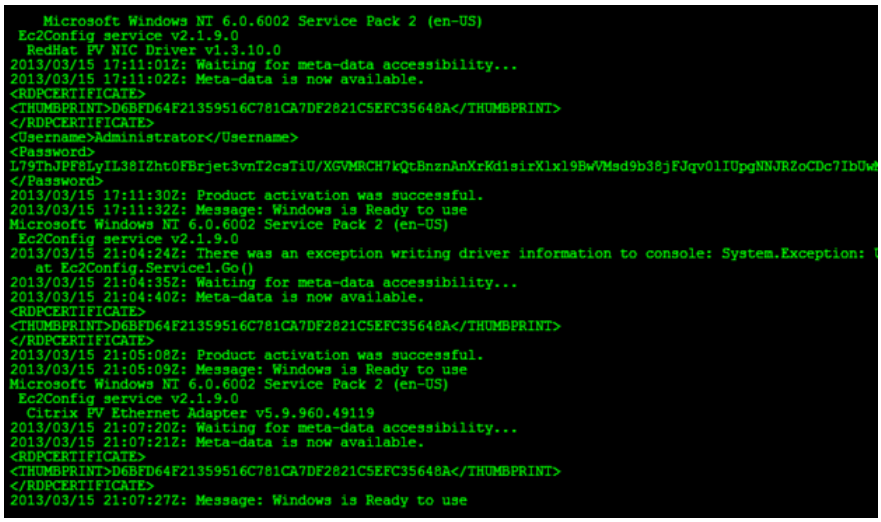
To upgrade a Windows Server 2008 or Windows Server 2008 R2 AMI

1. Connect to your instance and log in as the local administrator. For more information about connecting to your instance, see Connecting to Your Windows Instance Using RDP (p. 216).
2. In your instance, download the Citrix upgrade package by going to Amazon EC2 Windows Paravirtual Driver Upgrade Script.
3. Extract the contents of the upgrade package to a location of your choice.
4. Double-click the **Upgrade.bat** file. If you get a security warning, click **Run**.
5. In the **Upgrade Drivers** dialog box, review the information and click **Yes** if you are ready to start the upgrade.
6. In the **Red Hat Paravirtualized Xen Drivers for Windows ® uninstaller** dialog box, click **Yes** to remove the RedHat software. Your instance will be rebooted.

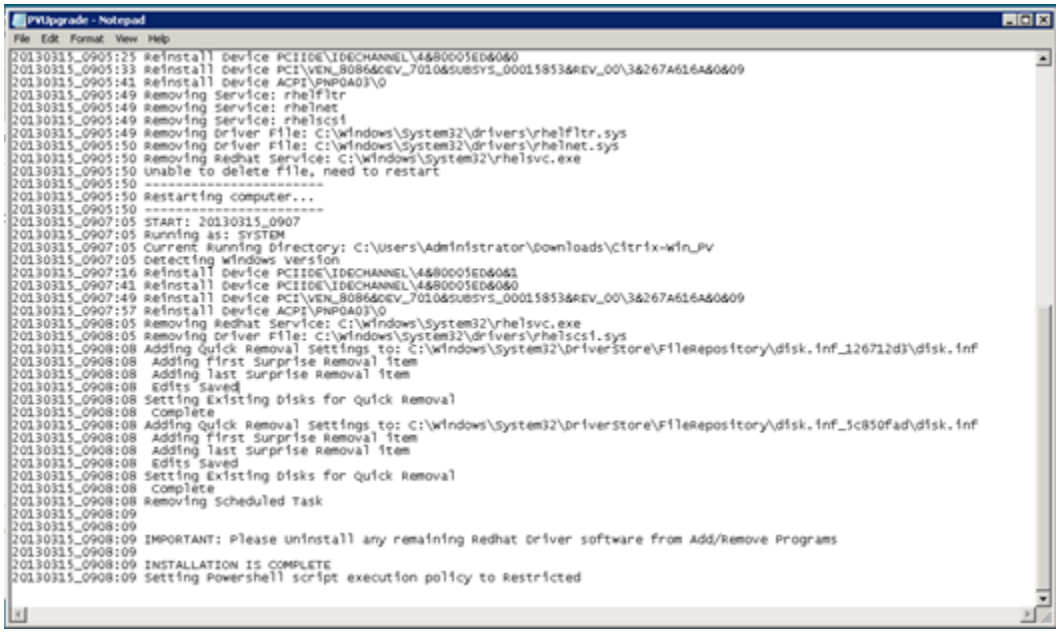
   Note
   If you do not see the uninstaller dialog box, click **Red Hat Paravirtualize…** in the Windows taskbar.

7. Check that the instance has rebooted and is ready to be used.

   a. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
   b. On the **Instances** page, right-click your instance and select **Get System Log**.
   c. The upgrade operations should have restarted the server 3 or 4 times. You can see this in the log file by the number of times `Windows is Ready to use` is displayed.



8. Connect to your instance and log in as the local administrator.
9. Close the **Red Hat Paravirtualized Xen Drivers for Windows ® uninstaller** dialog box.
10. Confirm that the installation is complete. Navigate to the `Citrix-WIN_PV` folder that you extracted earlier, open the `PVUpgrade.log` file, and then check for the text `INSTALLATION IS COMPLETE`.



Upgrading Your Citrix Xen Guest Agent Service

If you are using Citrix drivers on your Windows server, you can upgrade the Citrix Xen guest agent service. This Windows service handles tasks such as time synchronization at boot, as well as shutdown and restart events from the API. You can run this upgrade package on any version of Windows Server, including Windows Server 2012.

Before you start upgrading your drivers, make sure you back up your important information on the instance, or create an AMI from the instance. For more information about creating an AMI, see Creating an Amazon EBS-Backed Windows AMI (p. 68). If you create an AMI, make sure you do the following:

- Do not enable the Sysprep tool in the EC2Config service.
- Write down your password.
- Set your Ethernet adapter to DHCP.

To upgrade your Citrix Xen guest agent service

1. Connect to your instance and log in as the local administrator. For more information about connecting to your instance, see Connecting to Your Windows Instance Using RDP (p. 216).
2. In your instance, download the Citrix upgrade package by going to Amazon EC2 Windows Paravirtual Driver Upgrade Script.
3. Extract the contents of the upgrade package to a location of your choice.
4. Double-click the **Upgrade.bat** file. If you get a security warning, click **Run**.
5. In the **Upgrade Drivers** dialog box, review the information and click **Yes** if you are ready to start the upgrade.
6. When the upgrade is complete, the `PVUpgrade.log` file will open and contain the text `UPGRADE IS COMPLETE`.
7. Reboot your instance.

Upgrading PV Drivers on Your Windows Server 2003 Instance

Before you start upgrading your RedHat drivers to Citrix drivers, make sure you do the following:

- Back up your important information on the instance, or create an AMI from the instance. For more information about creating an AMI, see Creating an Amazon EBS-Backed Windows AMI (p. 68). If you create an AMI, make sure you do the following:
  - Do not enable the Sysprep tool in the EC2Config service.
  - Write down your password.
  - Set your Ethernet adapter to DHCP.
- Install the latest version of EC2Config by going to Amazon Windows EC2Config Service. For more information about the EC2Config service, see Configuring a Windows Instance Using the EC2Config Service (p. 235).

To upgrade a Windows Server 2003 AMI

1. Connect to your instance and log in as the local administrator. For more information about connecting to your instance, see Connecting to Your Windows Instance Using RDP (p. 216).
2. In your instance, download the Citrix upgrade package by going to Amazon EC2 Windows Paravirtual Driver Upgrade Script.
3. Extract the contents of the upgrade package to a location of your choice.
4. Double-click the **Upgrade.bat** file. If you get a security warning, click **Run**.
5. In the **Upgrade Drivers** dialog box, review the information and click **Yes** if you're ready to start the upgrade.
6. In the **Red Hat Paravirtualized Xen Drivers for Windows ® uninstaller** dialog box, click **Yes** to remove the RedHat software. Your instance will be rebooted.

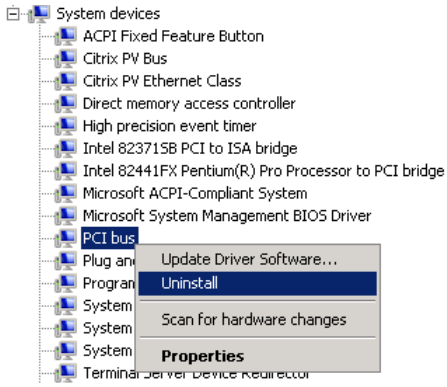
   Note
   If you do not see the uninstaller dialog box, click **Red Hat Paravirtualize…** in the Windows taskbar.

7. Check that the instance has been rebooted and is ready to be used.

   a. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
   b. On the **Instances** page, right-click your instance and select **Get System Log**.
   c. Check the end of the log message. It should read `Windows is Ready to use`.

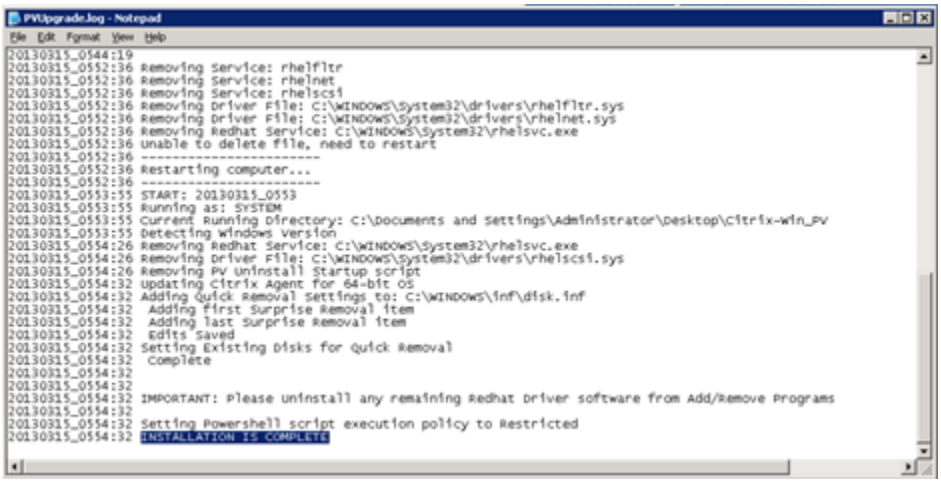
8. Connect to your instance and log in as the local administrator. The upgrade will continue by opening four applications: PowerShell, RedHat uninstaller, PVUpgrade.log and the Windows Device Manager.
9. Uninstall the PCI BUS.

   a. In the **Device Manager** window, expand **System devices**, right-click **PCI bus** and click **Uninstall**.



   b. When prompted, click **OK**.
   c. In the **System Settings Change** dialog, click **No** as you do not want to restart your instance immediately.
   d. Close **Device Manager**. The upgrade script reboots your instance.

10. Check that the instance is ready by repeating the procedure in step 7. After you've confirmed it is ready, log in as the administrator.
11. Confirm that the installation is complete. Navigate to the `Citrix-WIN_PV` folder that you extracted earlier, open the `PVUpgrade.log` file, and then check for the text `INSTALLATION IS COMPLETE`.



Troubleshooting PV Drivers

This topic describes solutions to common issues that you might encounter with Amazon EC2 PV drivers.

Contents

- Windows Server 2012 R2 loses network and storage connectivity after an instance reboot (p. 270)
- TCP Offloading (p. 274)
- Time Synchronization (p. 276)

Windows Server 2012 R2 loses network and storage connectivity after an instance reboot

Windows Server 2012 R2 Amazon Machine Images (AMIs) made available *before* September 10, 2014 can lose network and storage connectivity after an instance reboot. The error in the AWS Management Console system log states: “Difficulty detecting PV driver details for Console Output.” The connectivity loss is caused by the Windows Server 2012 R2 Plug and Play Cleanup feature. This features scans for and disables inactive system devices every 30 days. The feature incorrectly identifies the EC2 network device as inactive and removes it from the system. When this happens, the instance loses network connectivity after a reboot.

For systems that you suspect could be affected by this issue, you can download and run an in-place driver upgrade. If you are unable to perform the in-place driver upgrade, you can run a helper script. The script determines if your instance is affected. If it is affected, and the Amazon EC2 network device has *not* been removed, the script disables the Plug and Play Cleanup scan. If the Amazon EC2 network device has been removed, the script repairs the device, disables the Plug and Play Cleanup scan, and allows your instance to reboot with network connectivity enabled.

In this section

- Choose How You Want to Fix This Problem (p. 271)
- Method 1 - Enhanced Networking (p. 271)
- Method 2 - Registry configuration (p. 272)
- Run the Remediation Script (p. 274)

Choose How You Want to Fix This Problem

There are two methods for restoring network and storage connectivity to an instance affected by this issue. Choose one of the following methods:

Method	Prerequisites	Procedure Overview
Method 1 - Enhanced networking	Enhanced networking is only available in a virtual private cloud (VPC) which requires a C3 instance type. If the server does not currently use the C3 instance type, then you must temporarily change it. Enhanced networking is not available for ec2-classic.	You change the server instance type to a C3 instance. Enhanced networking then enables you to connect to the affected instance and fix the problem. After you fix the problem, you change the instance back to the original instance type. This method is typically faster than Method 2 and less likely to result in user error. You will incur additional charges as long as the C3 instance is running.
Method 2 - Registry configuration	Ability to create or access a second server. Ability to change Registry settings.	You detach the root volume from the affected instance, attach it to a different instance, connect, and make changes in the Registry. You will incur additional charges as long as the additional server is running. This method is slower than Method 1, but this method has worked in situations where Method 1 failed to resolve the problem.

Method 1 - Enhanced Networking

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Locate the affected instance. Open the context (right-click) menu for the instance, choose **Instance State**, and then choose **Stop**.

   Warning
   When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.
4. After the instance is stopped create a backup. Open the context (right-click) menu for the instance, choose **Image**, and then choose **Create Image**.
5. Change the instance type to any C3 instance type.
6. Start the instance.
7. Connect to the instance using Remote Desktop and then download https://s3.amazonaws.com/ec2-downloads-windows/Drivers/AWSPVDriverSetup.zip to the instance.
8. Extract the contents of the folder and run `AWSPVDriverSetup.msi`.

   After running the MSI, the instance automatically reboots and then upgrades the drivers. The instance will not be available for up to 15 minutes.
9. After the upgrade is complete and the instance passes both health checks in the Amazon EC2 console, connect to the instance using Remote Desktop and verify that the new drivers were installed.

In Device Manager, under **Storage Controllers**, locate **AWS PV Storage Host Adapter**. Check the driver properties for version 7.3.2.

10. Stop the instance and change the instance back to its original instance type.
11. Start the instance and resume normal use.

Method 2 - Registry configuration

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Locate the affected instance. Open the context (right-click) menu for the instance, choose **Instance State**, and then choose **Stop**.

   Warning
   When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

4. Choose **Launch Instance** and create a temporary Windows Server 2008 or Windows Server 2012 instance in the same Availability Zone as the affected instance. Do *not* create a Windows Server 2012 R2 instance.

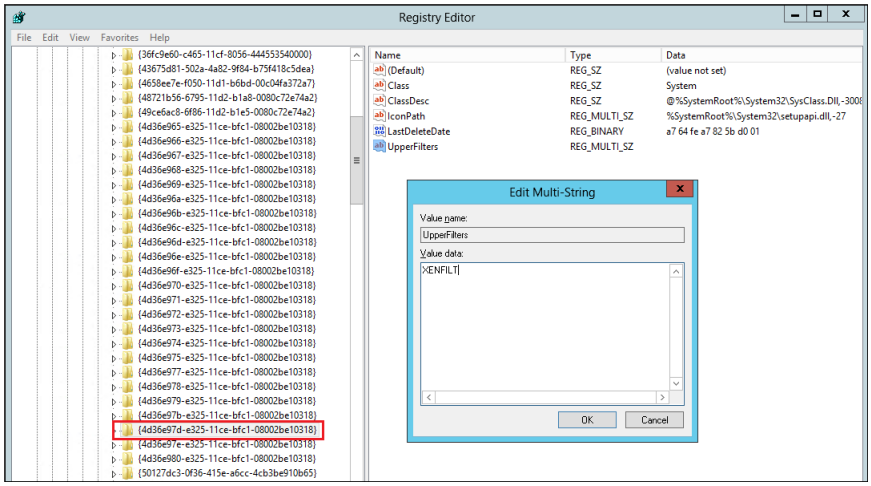
   Important
   If you do not create the instance in the same Availability Zone as the affected instance you will not be able to attach the root volume of the affected instance to the new instance.

5. In the navigation pane, choose **Volumes**.
6. Locate the root volume of the affected instance. Detach the volume and attach it to the temporary instance you created earlier. Attach it with the default device name (xvdf).
7. Use Remote Desktop to connect to the temporary instance, and then use the Disk Management utility to make the volume available for use.
8. On the temporary instance, open the Run dialog box, type regedit, and press Enter.
9. In the Registry Editor navigation pane, choose **HKEY_Local_Machine**, and then from the **File** menu choose **Load Hive**.
10. In the **Load Hive** dialog box, navigate to *Affected Volume*\Windows\System32\config\System and type a temporary name in the **Key Name** dialog box. For example, enter OldSys.
11. In the navigation pane of the Registry Editor, locate the following keys:

    HKEY_LOCAL_MACHINE*your_temporary_key_name*\ControlSet001\Control\Class\4d36e97d-e325-11ce-bfc1-08002be10318

    HKEY_LOCAL_MACHINE*your_temporary_key_name*\ControlSet001\Control\Class\4d36e96a-e325-11ce-bfc1-08002be10318

12. For each key, double-click **UpperFilters**, enter a value of XENFILT, and then click **OK**.



13. Locate the following key:

    HKEY_LOCAL_MACHINE*your_temporary_key_name*\ControlSet001\Services\XENBUS\Parameters
14. Create a new string (REG_SZ) with the name ActiveDevice and the following value:

    PCI\VEN_5853&DEV_0001&SUBSYS_00015853&REV_01
15. Locate the following key:

    HKEY_LOCAL_MACHINE*your_temporary_key_name*\ControlSet001\Services\XENBUS
16. Change the **Count** from 0 to 1.
17. Locate and delete the following keys:

    HKEY_LOCAL_MACHINE*your_temporary_key_name*\ControlSet001\Services\xenvbd\StartOverride

    HKEY_LOCAL_MACHINE *your_temporary_key_name*\ControlSet001\Services\xenfilt\StartOverride
18. In the Registry Editor navigation pane, choose the temporary key that you created when you first opened the Registry Editor.
19. From the **File** menu, choose **Unload Hive**.
20. In the Disk Management Utility, choose the drive you attached earlier, open the context (right-click) menu, and choose **Offline**.
21. In the Amazon EC2 console, detach the affected volume from the temporary instance and reattach it to your Windows Server 2012 R2 instance with the device name /dev/sda1. You must specify this device name to designate the volume as a root volume.
22. Start the instance.
23. Connect to the instance using Remote Desktop and then download https://s3.amazonaws.com/ec2-downloads-windows/Drivers/AWSPVDriverSetup.zip to the instance.
24. Extract the contents of the folder and run `AWSPVDriverSetup.msi`.

    After running the MSI, the instance automatically reboots and then upgrades the drivers. The instance will not be available for up to 15 minutes.
25. After the upgrade is complete and the instance passes both health checks in the Amazon EC2 console, connect to the instance using Remote Desktop and verify that the new drivers were installed. In Device Manager, under **Storage Controllers**, locate **AWS PV Storage Host Adapter**. Check the driver properties for version 7.3.2.
26. Delete or stop the temporary instance you created in this procedure.

Run the Remediation Script

If you are unable to perform an in-place driver upgrade or migrate to a newer instance you can run the remediation script to fix the problems caused by the Plug and Play Cleanup task.

To run the remediation script

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Choose the instance for which you want to run the remediation script. Open the context (right-click) menu for the instance, choose **Instance State**, and then choose **Stop**.

   Warning
   When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.
4. After the instance is stopped create a backup. Open the context (right-click) menu for the instance, choose **Image**, and then choose **Create Image**.
5. Open the context (right-click) menu for the instance, choose **Instance State**, and then choose **Start**.
6. Connect to the instance by using Remote Desktop and then download the RemediateDriverIssue.zip folder to the instance.
7. Extract the contents of the folder.
8. Run the remediation script according to the instructions in the Readme.txt file. The file is located in the folder where you extracted RemediateDriverIssue.zip.

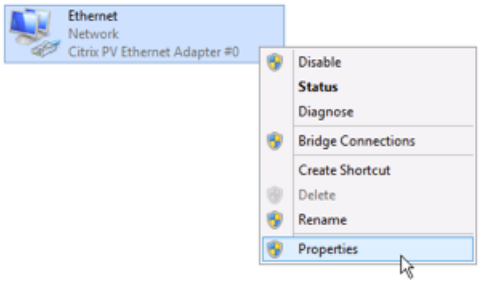
TCP Offloading

By default, TCP offloading is enabled for the Citrix PV drivers in Windows AMIs. If you encounter transport-level errors or packet transmission errors (as visible on the Windows Performance Monitor)—for example, when you're running certain SQL workloads—you may need to disable this feature.

Note
Disabling TCP offloading may reduce the network performance of your instance.

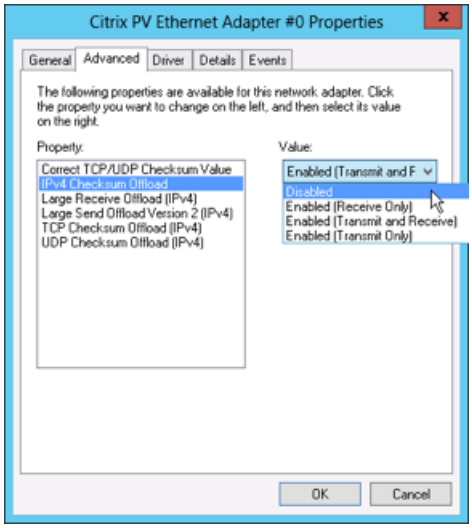
To disable TCP offloading for Windows Server 2012 and 2008

1. Connect to your instance and log in as the local administrator.
2. If you're using Windows Server 2012, press **Ctrl+Esc** to access the **Start** screen, and then click **Control Panel**. If you're using Windows Server 2008, click **Start** and select **Control Panel**.
3. Click **Network and Internet**, then **Network and Sharing Center**.
4. Click **Change adapter settings**.
5. Right-click **Citrix PV Ethernet Adapter #0** and select **Properties**.



6. In the **Local Area Connection Properties** dialog box, click **Configure** to open the **Citrix PV Ethernet Adapter #0 Properties** dialog box.
7. On the **Advanced** tab, disable each of the following properties by selecting them in the **Property** list, and selecting **Disabled** from the **Value** list:

   - **IPv4 Checksum Offload**
   - **Large Receive Offload (IPv4)**
   - **Large Send Offload Version 2 (IPv4)**
   - **TCP Checksum Offload (IPv4)**
   - **UDP Checksum Offload (IPv4)**



8. Click **OK**.
9. Run the following commands from a Command Prompt window.

```
C:\> netsh int ip set global taskoffload=disabled
C:\> netsh int tcp set global chimney=disabled
C:\> netsh int tcp set global rss=disabled
C:\> netsh int tcp set global netdma=disabled
```

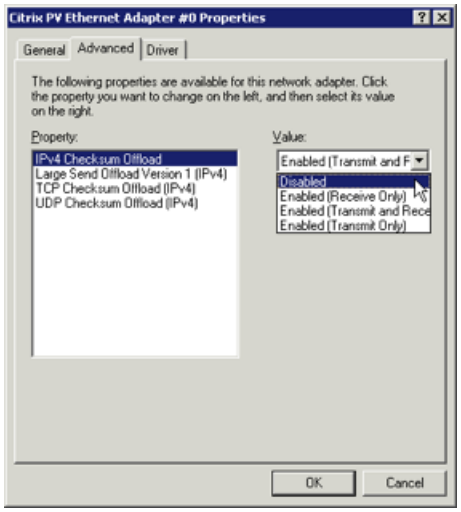
10. Reboot the instance.

To disable TCP offloading for Windows Server 2003

1. Connect to your instance and log in as the local administrator.
2. Click **Start**, and select **Control Panel**, then **Network Connections**, and then **Local Area Connection 3**.
3. Click **Properties**.
4. In the **Local Area Connection 3** dialog box, click **Configure...** to open the **Citrix PV Ethernet Adapter #0 Properties** dialog box.
5. On the **Advanced** tab, disable each of the following properties by selecting them in the **Property** list, and selecting **Disabled** from the **Value** list:

   - **IPv4 Checksum Offload**
   - **Large Send Offload Version 1 (IPv4)**

- **TCP Checksum Offload (IPv4)**
- **UDP Checksum Offload (IPv4)**



6. Click **OK**.
7. Run the following PowerShell script.

```
$n = Get-ItemProperty "HKLM:\SYSTEM\Select" | Select -expand Current

$root = "HKEY_LOCAL_MACHINE\SYSTEM\ControlSet00$n\Control\Class\{4D36E972-
E325-11CE-BFC1-08002BE10318}"
$items = Get-ChildItem -Path Registry::$Root -Name

Foreach ($item in $items) {
    if ($item -ne "Properties") {
      $path = $root + "\" + $item
      $DriverDesc = Get-ItemProperty -Path Registry::$path | Select-Object
 -expandproperty DriverDesc
      if ($DriverDesc -eq "Citrix PV Ethernet Adapter") {
         Set-ItemProperty -path Registry::$path -Name *IPChecksumOffloadIPv4
 -Value 0
         Set-ItemProperty -path Registry::$path -Name *TCPChecksumOffloadIPv4
 -Value 0
         Set-ItemProperty -path Registry::$path -Name *UDPChecksumOffloadIPv4
 -Value 0
          Set-ItemProperty -path Registry::$path -Name *LSOv1IPv4 -Value 0
      }
    }
}
```

8. Reboot the instance.

Time Synchronization

Prior to the release of the 2013.02.13 Windows AMI, the Citrix Xen guest agent could set the system time incorrectly. This can cause your DHCP lease to expire. If you have issues connecting to your instance, you might need to update the agent.

To determine whether you have the updated Citrix Xen guest agent, check whether the `C:\Program Files\Citrix\XenGuestAgent.exe` file is from March 2013. If the date on this file is earlier than that, update the Citrix Xen guest agent service. For more information, see Upgrading Your Citrix Xen Guest Agent Service (p. 267).

Related Topics

For information about troubleshooting EC2 Windows instances, see Troubleshooting Windows Instances (p. 687).

Setting Passwords for Windows Instances

When you connect to a Windows instance, you must specify a user account that has permission to access the instance, along with the password for the account. The first time that you connect to your instance, specify the Administrator account and the default password. This default password is automatically generated by the EC2Config service.

After you connect to your instance, we recommend that you change the Administrator password from its default value. If you lose your password or it expires, you can manually configure EC2Config to generate a new password.

Contents

- Changing the Administrator Password After Connecting (p. 277)
- Resetting an Administrator Password that's Lost or Expired (p. 278)

Changing the Administrator Password After Connecting

Use the following procedure to change the password for the Administrator account for your instance.

> **Important**
> Store the new password in a safe place, because you can't get it using the Amazon EC2 console; the console always gets the default password. If you attempt to connect to the instance using the default password after the password was changed, you'll get the error "Your credentials did not work."

To change the local Administrator password

1. Connect to your instance.
2. From your instance, open a Command Prompt window.
3. From the Command Prompt window, run the following command:

```
C:\> net user Administrator new_password
```

Resetting an Administrator Password that's Lost or Expired

If you've lost the password for the local Administrator account for your Windows instance, or if the password has expired, you can reset the password using the EC2Config service. Note that you can't reset the password if you've disabled the local Administrator account.

You'll use the EC2Config service to reset the administrator password by modifying one of its configuration files on the boot volume of the instance that needs the password reset. However, this file can't be modified unless the volume is not currently the root volume. Therefore, you must detach the root volume from the instance, attach the volume to another instance as a secondary volume, change the configuration settings, and then reattach the volume as the root volume.

> **Important**
> The instance gets a new public IP address after you stop and start it as described in the following procedure. After resetting the password, be sure to connect to the instance using its current public DNS name. If the instance is in EC2-Classic, any Elastic IP address is disassociated from the instance, so you must reassociate it. For more information, see Instance Lifecycle (p. 203).

To reset the Administrator password

1. Verify that the EC2Config service is installed on the instance that needs a password reset. (This instance is referred to as the *original instance* in this procedure.) EC2Config is available by default on all Amazon Windows AMIs, or you can download it. For more information, see Installing the Latest Version of EC2Config (p. 256).
2. Open the Amazon EC2 console.
3. Stop the original instance as follows:
   a. In the navigation pane, click **Instances**.
   b. Right-click the original instance and then click **Stop**.

      > **Warning**
      > When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

   c. In the **Stop Instances** dialog box, click **Yes, Stop**. After the instance has stopped, proceed with the next step.
4. Launch a Windows instance in the same Availability Zone as the original instance. (This instance is referred to as the *temporary instance* in this procedure.)

   > **Warning**
   > If your temporary instance is based on the same AMI that the original instance is based on, and the operating system is later than Windows Server 2003, you must complete additional steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision. Alternatively, select a different AMI for the temporary instance. For example, if the original instance uses the AWS Windows AMI for Windows Server 2008 R2, launch the temporary instance using the AWS Windows AMI for Windows Server 2012 or Windows Server 2003. (To find an AMI for Windows Server 2003, search for an AMI using the name `Windows_Server-2003-R2_SP2`.)
5. Detach the root volume from the original instance as follows:
   a. On the **Description** pane of the original instance, note the volume ID of the volume listed as the **Root device**.
   b. In the navigation pane, click **Volumes**.

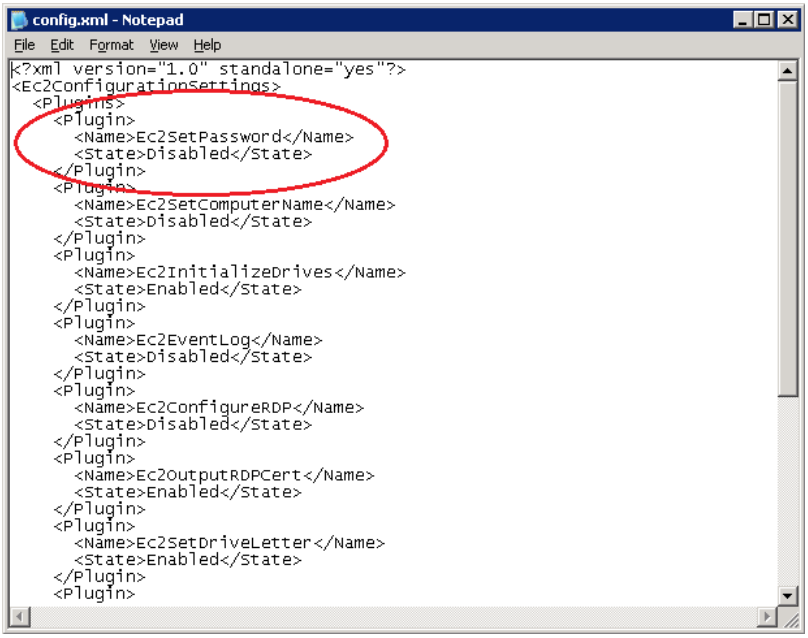
c. In the list of volumes, right-click the volume, and then click **Detach Volume**. After the volume's status changes to **available**, proceed with the next step.

6. Attach the volume to the temporary instance as a secondary volume as follows:

   a. Right-click the volume and click **Attach Volume**.
   b. In the **Attach Volume** dialog box, start typing the name or ID of your temporary instance in the **Instances** field, and then select it from the list of suggested options.
   c. In the **Device** box, type **xvdf** (if it isn't already there), and then click **Attach**.
   d. Connect to the temporary instance, open the **Disk Management** utility, and bring the drive online. For more information, see Making the Volume Available on Windows (p. 530).

7. Modify the configuration file on the secondary volume as follows:

   a. From the temporary instance, open `\Program Files\Amazon\Ec2ConfigService\Settings\config.xml` using a text editor, such as Notepad.
   b. At the top of the file, find the plugin with the name `Ec2SetPassword`, as shown here. Change the state from `Disabled` to `Enabled` and then save the file.



8. (Optional) If your temporary instance is based on the same AMI that the original instance is based on, and the operating system is later than Windows Server 2003, you must complete the following steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision.

   a. In the Registry Editor, load the following registry hive into a folder named `BCD`: `d:\boot\bcd`.
   b. Search for the following data value in BCD: "Windows Boot Manager". You'll find a match under a key named `12000004`.
   c. Select the key named `11000001` that is sibling to the key you found in the previous step. View the data for the `Element` value.

d. Locate the four-byte disk signature at offset 0x38 in the data. Reverse the bytes to create the disk signature, and write it down. For example, the disk signature represented by the following data is E9EB3AA5:

```
...
0030  00 00 00 00 01 00 00 00
0038  A5 3A EB E9 00 00 00 00
0040  00 00 00 00 00 00 00 00
...
```

e. In a Command Prompt window, run the following command to start Microsoft DiskPart.

```
C:\> diskpart
```

f. Run the following DiskPart command to select the volume. (You can verify that the disk number is 1 using the **Disk Management** utility.)

```
DISKPART> select disk 1

Disk 1 is now the selected disk.
```

g. Run the following DiskPart command to get the disk signature.

```
DISKPART> uniqueid disk

Disk ID: 0C764FA8
```

h. If the disk signature shown in the previous step doesn't match the disk signature from BCD that you wrote down earlier, use the following DiskPart command to change the disk signature so that it matches:

```
DISKPART> uniqueid disk id=E9EB3AA5
```

9. Detach the secondary volume from the temporary instance as follows:

   a. Using the **Disk Management** utility, bring the volume offline.

      Note
      The drive is automatically offline if the temporary instance is running the same operating system as the affected instance, so you won't need to bring it offline manually.

   b. From the Amazon EC2 console, in the navigation pane, click **Volumes**.
   c. In the list of volumes, right-click the volume, and then click **Detach Volume**. After the volume's status changes to **available**, proceed with the next step.

10. Reattach the volume to the original instance as its root volume as follows:

    a. Right-click the volume and then click **Attach Volume**.
    b. In the **Attach Volume** dialog box, start typing the name or ID of the original instance in the **Instances** list, and then select the instance.
    c. In the **Device** box, enter `/dev/sda1`.

d. Click **Yes, Attach**.

11. Restart the original instance as follows:

    a. In the navigation pane, click **Instances**.
    b. Right-click the original instance and then click **Start**.
    c. In the **Start Instances** dialog box, click **Yes, Start**.

12. Retrieve the new default password as follows:

    a. In the navigation pane, click **Instances**.
    b. Right-click the original instance and then click **Get Windows Password**.
    c. In the **Retrieve Default Windows Administrator Password** dialog box, click **Browse**, and then select the `.pem` file that corresponds to the key pair that you specified when you launched the instance.
    d. Click **Decrypt Password**. You'll use the decrypted password to connect to the original instance using the local Administrator account.

Setting the Time for a Windows Instance

A consistent and accurate time reference is crucial for many server tasks and processes. Most system logs include a time stamp that you can use to determine when problems occur and in what order the events take place. If you use the AWS CLI, EC2 CLI, or an AWS SDK to make requests from your instance, these tools sign requests on your behalf. If your instance's date and time are not set correctly, the date in the signature may not match the date of the request, and AWS rejects the request. We recommend that you use Coordinated Universal Time (UTC) for your Windows instances. However, you can use a different time zone if you want.

Contents

- Changing the Time Zone (p. 281)
- Configuring Network Time Protocol (NTP) (p. 282)
- Configuring Time Settings for Windows Server 2008 and later (p. 283)
- Configuring Time Settings for Windows Server 2003 (p. 284)
- Related Topics (p. 284)

Changing the Time Zone

Windows instances are set to the UTC time zone by default. you can change the time to correspond to your local time zone or a time zone for another part of your network.

To change the time zone on an instance

1. From your instance, open a Command Prompt window.
2. Identify the time zone to use on the instance. To get a list of time zones, use the following command: **tzutil /l**. This command returns a list of all available time zones, using the following format:

```
display name
time zone ID
```

3. Locate the time zone ID to assign to the instance.
4. Assign the time zone to the instance by using the following command:

```
C:\> tzutil /s "Pacific Standard Time"
```

The new time zone should take effect immediately.

Configuring Network Time Protocol (NTP)

Windows instances use the time.windows.com NTP server to configure the system time; however, you can change the instance to use a different set of NTP servers if you need to. For example, if you have Windows instances that do not have Internet access, you can configure them to use an NTP server located within your private network. Your instance's security group must allow outbound UDP traffic on port 123 (NTP). The procedures in this section show how you can verify and change the NTP configuration for an instance.

To verify the NTP configuration

1. From your instance, open a Command Prompt window.
2. Get the current NTP configuration by typing the following command:

```
C:\> w32tm /query /configuration
```

This command returns the current configuration settings for the Windows instance.

3. (Optional) Get the status of the current configuration by typing the following command:

```
C:\> w32tm /query /status
```

This command returns information such as the last time the instance synced with the NTP server and the poll interval.

To change the NTP configuration

1. From the Command Prompt window, run the following command:

```
C:\> w32tm /config /manualpeerlist:comma-delimited list of NTP servers
/syncfromflags:manual /update
```

Where *comma-delimited list of NTP servers* is the list of NTP servers for the instance to use.

2. Verify your new settings by using the following command:

```
C:\> w32tm /query /configuration
```

Configuring Time Settings for Windows Server 2008 and later

When you change the time on a Windows instance, you must ensure that the time persists through system restarts. Otherwise, when the instance restarts, it reverts back to using UTC time. For Windows Server 2008 and later, you can persist your time setting by adding a **RealTimeIsUniversal** registry key.

To set the RealTimeIsUniversal registry key

1. From the instance, open a Command Prompt window.
2. Use the following command to add the registry key:

```
C:\> reg add "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformation" /v RealTimeIsUniversal /d 1 /t REG_DWORD /f
```

3. (Optional) If you are using an AMI that was created before February 22, 2013, you should verify that the Microsoft hotfix KB2800213 is installed. If this hotfix is not installed, install it. This hotfix resolves a known issue in which the **RealTimeIsUniversal** key causes the Windows CPU to run at 100% during Daylight savings events and the start of each calendar year (January 1).

   If you are using an AMI running Windows Server 2008 R2, you must verify that the Microsoft hotfix KB2922223 is installed. If this hotfix is not installed, install it. This hotfix resolves a known issue in which the **RealTimeIsUniversal** key prevents the system from updating the CMOS clock.
4. (Optional) Verify that the instance saved the key successfully using the following command:

```
C:\> reg query "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformation" /s
```

   This command returns the subkeys for the **TimeZoneInformation** registry key. You should see the **RealTimeIsUniversal** key at the bottom of the list, similar to the following:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformation
    Bias                            REG_DWORD     0x1e0
    DaylightBias                    REG_DWORD     0xffffffc4
    DaylightName                    REG_SZ        @tzres.dll,-211
    DaylightStart                   REG_BINARY
00000300020002000000000000000000
    StandardBias                    REG_DWORD     0x0
    StandardName                    REG_SZ        @tzres.dll,-212
    StandardStart                   REG_BINARY
00000B0001000200000000000000000
    TimeZoneKeyName                 REG_SZ        Pacific Standard Time
    DynamicDaylightTimeDisabled     REG_DWORD     0x0
    ActiveTimeBias                  REG_DWORD     0x1a4
    RealTimeIsUniversal             REG_DWORD     0x1
```

Configuring Time Settings for Windows Server 2003

When you change the time zone on an instance running Windows Server 2003, you must ensure that the time persists through system restarts. Otherwise, if you restart the instance, it reverts to using the UTC clock for your time zone, resulting in a time skew that correlates with your time offset. You can persist your time setting by updating your Citrix PV drivers. For more information, see Upgrading PV Drivers on Your Windows AMI (p. 265).

After you update the Citrix PV drivers, the Citrix Tools for Virtual Machines Service sets the time on the instance when the service is started.

Related Topics

For more information about how the Windows operating system coordinates and manages time, including the addition of a leap second, see the following topics:

- How the Windows Time Service Works (TechNet)
- W32tm (TechNet)
- How the Windows Time service treats a leap second (TechNet)
- The story around Leap Seconds and Windows: It's likely not Y2K (blog)

Managing Windows Instance Configuration

The Amazon EC2 Simple Systems Manager (SSM) feature enables you to manage the configuration of your Windows instances while they are running. You create a *configuration document*, which describes configuration tasks (for example, installing software), and then associate the configuration document with one or more running Windows instances. The configuration agent on the instance processes the configuration document and configures the instance as specified.

If you disassociate a configuration document from an instance, this doesn't change the configuration of the instance. To change the configuration of an instance after you disassociate a configuration document, you must create a new configuration document that describes the configuration tasks (for example, uninstalling software), and then associate it with the instance.

To run scripts at instance launch only, consider using user data execution instead. For more information, see Executing User Data (p. 243).

For more complex automation scenarios, consider using AWS CloudFormation or AWS OpsWorks instead. For more information, see the *AWS CloudFormation User Guide* or the *AWS OpsWorks User Guide*.

Prerequisites

The EC2Config service processes SSM configuration documents and configures the instance as specified. Download the latest version of Amazon Windows EC2Config Service to each server you want to configure with SSM.

Limitations

- SSM is supported only for Windows instances.
- SSM is supported only in the following regions:
  - US East (N. Virginia) region (`us-east-1`).
  - US West (Oregon) region (`us-west-2`).

- EU (Ireland) region (`eu-west-1`).

To manage the configuration of your Windows instances using SSM, complete the following tasks.

Tasks

- Grant IAM Users Access to SSM (p. 285)
- Prepare the Instance (p. 285)
- Create the JSON File (p. 286)
- Create the Configuration Document (p. 288)
- Associate the Configuration Document with the Instance (p. 288)
- Manually Apply the Configuration (p. 289)
- Disassociate the Configuration Document from the Instance (p. 289)
- Delete the Configuration Document (p. 289)
- Troubleshooting (p. 290)

Grant IAM Users Access to SSM

To allow an IAM user to use SSM to configure an instance, you must grant the user permission to use the actions specified in the following example policy document:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ssm:*",
      "Resource": "*"
    }
  ]
}
```

Prepare the Instance

To configure an instance using SSM, you must launch it with an IAM role that grants permission to use the SSM API, and ensure the instance has the latest version of the EC2Config service.

Granting Permission to use the SSM API

If you launched your Windows instance using an IAM role, you can add a policy that grants permission to use the SSM API. Otherwise, you must launch a new instance with an IAM role. For information about creating an IAM role, see Creating a Role to Delegate Permissions to an AWS Service in the *IAM User Guide*.

The following is an example of a policy document that grants SSM permission to configure an instance on your behalf:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessToSSM",
      "Effect": "Allow",
```

```
      "Action": [
        "ssm:DescribeAssociation",
        "ssm:ListAssociations",
        "ssm:GetDocument",
        "ssm:UpdateAssociationStatus",
        "ds:CreateComputer"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

When you create an IAM role, you must associate a trust policy with the role to ensure that Amazon EC2 is allowed to assume the role.

To edit a trust policy

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**, and then choose the name of your role.
3. Under **Trust Relationships**, choose **Edit Trust Relationship**.
4. Edit the trust policy as needed, and then choose **Update Trust Policy**. The following is an example of a trust policy that allows Amazon EC2 to assume the role.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {"Service": "ec2.amazonaws.com"},
    "Action": "sts:AssumeRole"
  }
}
```

Installing the Latest EC2Config

Using SSM requires EC2Config version 3.0 or later. If you launched your instance from a current Windows AMI, then it has the latest version of EC2Config. Otherwise, you can install the latest version of EC2Config on your instance. For more information, see Installing the Latest Version of EC2Config (p. 256).

If you launch Windows instances using your own AMI, you should create a new AMI after you install EC2Config on your instance. Consider installing other updates on your instance before you create your AMI. For more information, see Updating Your Windows Instance (p. 79) and Creating an Amazon EBS-Backed Windows AMI (p. 68).

Create the JSON File

Open a text editor, add the JSON to describe the configuration, and then save the file with a `.json` file extension.

For more information about the structure of the JSON for a configuration document, see Configuration Document in the *Amazon EC2 Simple Systems Manager API Reference*.

Example: Install Applications

The following JSON describes applications to install on the instance. For each application, `source` is the URL of its `.msi` file.

```
{
  "schemaVersion": "1.0",
  "description": "Example instance configuration tasks",
  "runtimeConfig": {
    "aws:applications": {
      "properties": [
        {
          "action": "Install",
          "source": "http://dev.mysql.com/get/Downloads/MySQLInstaller/mysql-
installer-community-5.6.22.0.msi"
        },
        {
          "action": "Install",
          "source": "https://www.python.org/ftp/python/2.7.9/python-2.7.9.msi"

        },
        {
          "action": "Install",
          "source": "http://download.winzip.com/winzip190-64.msi",
          "parameters": "INSTALLDIR=\"C:\\Program Files\\WinZipXX\""
        }
      ]
    }
  }
}
```

Example: Install PowerShell Modules and Run Commands

The following JSON describes PowerShell modules to install on your instance. For each module, `source` is the URL of the module and `runCommand` specifies the PowerShell command to run.

```
{
  "schemaVersion": "1.0",
  "description": "Example instance configuration tasks",
  "runtimeConfig": {
    "aws:psModule": {
      "properties": [
        {
         "description": "Example to install windows update PS module and install
 all .NET 4 updates.",
         "source": "https://gallery.technet.microsoft.com/scriptcenter/2d191bcd-
3308-4edd-9de2-88dff796b0bc/file/41459/43/PSWindowsUpdate.zip",
          "runCommand": "Get-WUInstall -ServiceID 9482f4b4-e343-43b6-b170-
9a65bc822c77 -Title \".NET Framework 4\" -AcceptAll"
        },
        {
          "description": "Example to install chocolatey package provider and
use it to install 7zip and GoogleChrome.",
          "runCommand": [
            "$url = 'https://chocolatey.org/install.ps1'" ,
            "iex ((new-object net.webclient).DownloadString($url))",
            "choco install -y 7zip",
            "choco install -y GoogleChrome"
          ]
```

```
        }
      ]
    }
  }
}
```

Example: Join an AWS Domain

For information about using SSM to join a Windows instance to a directory, see Joining a Windows Instance to an AWS Directory Service Domain (p. 291).

Example: Send Data to Amazon CloudWatch

For information about using SSM to send data to Amazon CloudWatch, see Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs (p. 299).

Create the Configuration Document

Use the AWS CLI or the Tools for Windows PowerShell to create a configuration document, specifying the JSON file that you created in the previous task.

AWS CLI

Use the following create-document command to name this configuration and make it available for use.

```
aws ssm create-document --content file://my-config.json --name "my-custom-config"
```

Tools for Windows PowerShell

Use the following New-SSMDocument command to name this configuration and make it available for use.

```
$doc = Get-Content my-config.json | Out-String
New-SSMDocument -Content $doc -Name "my-custom-config"
```

Associate the Configuration Document with the Instance

Use the AWS CLI or the Tools for Windows PowerShell to associate a configuration document with an instance. You'll specify the name of the configuration document that you created in the previous task. An instance can be associated with one configuration document at a time. If you associate a configuration document with an instance that already has an associated configuration document, the new configuration document replaces the existing configuration document.

AWS CLI

Use the following create-association command to associate your configuration document with your Windows instance.

```
aws ssm create-association --instance-id i-1a2b3c4d --name "my-custom-config"
```

Tools for Windows PowerShell

Use the following New-SSMAssociation command to associate your configuration document with your Windows instance.

```
New-SSMAssociation -InstanceId i-1a2b3c4d -Name "my-custom-config"
```

Manually Apply the Configuration

If you need to ensure that your instance is configured as specified in its current configuration document, you can run the `ec2config-cli` tool on your instance as follows:

```
ec2config-cli --apply-configuration
```

Alternatively, you can use Windows Task Scheduler to run `ec2config-cli` periodically to ensure that your instance maintains this configuration.

You can verify that `ec2config-cli` is installed by checking for it in the `C:\Program Files\Amazon\Ec2ConfigService` directory. If you do not have `ec2config-cli`, you can get it by installing the current version of the EC2Config service. For more information, see Installing the Latest Version of EC2Config (p. 256).

Disassociate the Configuration Document from the Instance

You can't update a configuration document after you create it. To associate a different configuration document with your instance, you can delete the existing association, and then associate a new configuration document with your instance. Note that terminating an instance does not automatically disassociate an associated configuration document.

AWS CLI

Use the following delete-association command to disassociate a configuration document from your Windows instance.

```
aws ssm delete-association --instance-id i-1a2b3c4d --name "my-custom-config"
```

Tools for Windows PowerShell

Use the following Remove-SSMAssociation command to disassociate a configuration document from your Windows instance.

```
Remove-SSMAssociation -InstanceId i-1a2b3c4d -Name "my-custom-config"
```

Delete the Configuration Document

When you are finished with a configuration document, you can delete it. You must disassociate the configuration document from any instances it is associated with before you can delete it.

AWS CLI

Use the following delete-document command to delete your configuration document.

```
aws ssm delete-document --name "my-custom-config"
```

Tools for Windows PowerShell

Use the following Remove-SSMDocument command to delete your configuration document.

```
Remove-SSMDocument -Name "my-custom-config"
```

Troubleshooting

This section includes information to help you troubleshoot problems with SSM.

Log4net Logging

The EC2Config service logs information in the following files using Apache log4net. The information in these files can help you troubleshoot problems.

- C:\Windows\System32\winevt\Logs\EC2ConfigService.evtx
- C:\Program Files\Amazon\Ec2ConfigService\Logs
- LocalSystem %LOCALAPPDATA%
  - **Windows Server 2008 or later**

    C:\Windows\System32\config\systemprofile\AppData\Amazon\Ec2Config\InstanceData\Logs\Ec2ConfigPluginFramework.txt
  - **Windows Server 2003**

    C:\Documents and Settings\Default User\Local Settings\Amazon\Ec2Config\InstanceData\Logs\Ec2ConfigPluginFramework.txt

You can enable extended logging by updating the log4net.config file. By default, the configuration file is located here:

C:\Program Files\Amazon\Ec2ConfigService\log4net.config

For more information about log4net configuration, see Apache log4net Manual - Configuration. For examples of log4net configurations, see Apache log4net Config Examples.

Windows Event Logs

The EC2Config service also logs information in a Windows Event log named *Ec2ConfigService*.

You can extract information from this event log to a log file by executing the following command from an elevated PowerShell command prompt:

```
Get-EventLog Ec2ConfigService | Sort-Object Index | Format-Table Message
-AutoSize -Wrap | Out-File -Width 240 "C:\Program
Files\Amazon\Ec2ConfigService\Logs\PluginFramework.txt"
```

If you want to log Windows Events to a log file with debugging enabled you must update the log4net.config file root element as follows:

```
<root> <level value="DEBUG"/> <appender-ref ref="RollingFileAppender"/>
<appender-ref ref="EventLogAppender"/> </root>
```

EC2 Console System Log

The following output in the EC2 console system log indicates that the EC2Config service was unable to connect to an SSM endpoint. These issues indicate problems with authorization and IAM role permissions, as noted in the following output messages:

```
 Info: EC2Config configuration status:3;region:us-east-1;iam:0;authz:0 The
output can
   help you troubleshoot the cause of the failure: configuration status:3: The
 calls to SSM
   failed. Ensure that you have granted the required IAM permissions to IAM
users. SSM also
   requires an Internet connection from your instance.
```

```
 iam:0: The instance was not launched with an IAM role. You cannot download
documents
   if there is no IAM role/credentials associated with the instance.
```

```
 authz:0: The instance is not authorized to access SSM. This happens if you
launched
  the instance without an IAM role, or if the role associated with your instance
 does not
   have the necessary permissions to access the service.
```

You can troubleshoot specific reasons for a configuration document execution failure by checking the status of the association using the describe-association (AWS CLI) command or the Get-SSMAssociation (Tools for Windows PowerShell) command.

Joining a Windows Instance to an AWS Directory Service Domain

You can join an Amazon EC2 Windows instance to an active AWS Directory Service directory or AD Connector directory using Amazon EC2 Simple Systems Manager (SSM). To perform this task with SSM, you use the AWS CLI or AWS Tools for Windows PowerShell to create an SSM configuration document that specifies the domain join details, and then associate the configuration document with a running instance.

Alternatively, you can use the launch instance wizard in the Amazon EC2 console to launch an instance and specify the domain that you want to join. The wizard searches for any existing configuration documents for the domain in your account to associate with your instance; if it can't locate one, it creates a configuration document for you, and immediately associates it with your running instance.

After you've associated the configuration document with your instance, you can connect to the instance using domain credentials you've defined in your AWS Directory Service directory.

There's no additional charge for using SSM or joining your instance to a domain. Standard charges for instance usage and AWS Directory Service usage apply.

For more information about SSM, see Managing Windows Instance Configuration (p. 284).

Contents

- Limitations (p. 292)
- Prerequisites (p. 292)
- Joining a Domain Using the AWS CLI or AWS Tools for Windows PowerShell (p. 293)

- Joining a Domain Using the Amazon EC2 Launch Wizard (p. 295)
- Getting the Domain Join Status (p. 296)
- Connecting To Your Instance Using Domain Credentials (p. 297)
- Troubleshooting (p. 297)
- Viewing Information About Your Associations (p. 298)
- Changing an Association (p. 299)
- Deleting a Configuration Document (p. 299)

Limitations

- SSM is supported only for Windows instances.
- SSM is supported only in the following regions:
  - US East (N. Virginia) region (`us-east-1`).
  - US West (Oregon) region (`us-west-2`).
  - EU (Ireland) region (`eu-west-1`).

  In other regions, you can manually join an instance to a domain. For more information, see Joining an Instance to an AWS Directory Service Directory in the *AWS Directory Service Administration Guide*.

Prerequisites

- To join a domain, ensure that you have the following resources available or configured in your AWS account:
  - An active AWS Directory Service directory. For more information about creating a directory, see Getting Started with AWS Directory Service in the *AWS Directory Service Administration Guide*.
    - To create a directory, you must have a VPC with two subnets. For more information about creating a VPC, see What is Amazon VPC? in the *Amazon VPC User Guide*. Instances that you join to the domain must be launched into the same VPC in which your domain is located.
  - A Windows instance that meets the requirements described in Prepare the Instance (p. 285).
  - An Internet connection for your instance, so that it can communicate with SSM. Ensure that you have a public subnet into which to launch your instance, and ensure that your instance has a public IP address. Alternatively, you can launch your instance into a private subnet without assigning it a public IP address, and use a NAT instance in a public subnet to initiate traffic to the Internet. For more information about NAT, see NAT Instances in the *Amazon VPC User Guide*.
- If you are using the AWS CLI or the AWS Tools for Windows PowerShell to create a configuration document, you need the following information:
  - The name and ID of the directory to join.
  - The IP addresses of the DNS servers in the AWS Directory Service directory. For more information, see Get the DNS Server Address in the *AWS Directory Service Administration Guide*.
- To allow an IAM user to use SSM to configure an instance, you must grant the user permission to use the actions specified in the following example policy document:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ssm:*",
      "Resource": "*"
```

```
    }
  ]
}
```

- Your instance must be launched with an IAM role that grants permission to use the SSM API. For more information, see Prepare the Instance (p. 285).

Joining a Domain Using the AWS CLI or AWS Tools for Windows PowerShell

To use the AWS CLI or the AWS Tools for Windows PowerShell to join a domain, you must create a configuration document, and then associate the configuration document with an already running instance.

To construct the configuration document, use a text editor of your choice, and save the file with the `*.json` extension. For more information about the structure of a configuration document, see Configuration Document in the *Amazon EC2 Simple Systems Manager API Reference*.

Use the following AWS CLI or AWS Tools for Windows PowerShell commands to create the configuration document, launch an instance, and then associate the file with your instance.

Action	AWS CLI	AWS Tools for Windows PowerShell
To create a configuration document in your account.	create-document	New-SSMDocument
To launch an instance. You can also join an existing instance to a domain, provided it meets the prerequisites. For more information, see Prerequisites (p. 292).	run-instances	New-EC2Instance
To associate the configuration document with your instance.	create-association	New-SSMAssociation

To join a domain using the AWS CLI or AWS Tools for Windows PowerShell

1. Open a text editor on your computer, and write a configuration document. When you are done, save the file with a `.json` extension. The following is an example of a configuration document that allows instances to join domain `d-1234567890`:

```
{
    "schemaVersion": "1.0",
    "description": "Sample configuration to join an instance to a domain",
    "runtimeConfig": {
        "aws:domainJoin": {
            "properties": {
                "directoryId": "d-1234567890",
                "directoryName": "test.example.com",
                "directoryOU": "OU=test,DC=example,DC=com",
                "dnsIpAddresses": [
                    "198.51.100.1",
                    "198.51.100.2"
```

```
                ]
            }
        }
    }

}
```

2. Create the configuration document in your account, and give it a name. The name of the file must be between 1 and 64 characters in length.

   AWS CLI

   ```
   aws ssm create-document --content file://path/to/myconfigfile.json --name
   "My_Custom_Config_File"
   ```

   Tools for Windows PowerShell

   First create a variable that contains the file contents, and then create the document.

   ```
   $doc = Get-Content C:\temp\myconfigfile.json | Out-String
   New-SSMDocument -Content $doc -Name "My_Custom_Config_File"
   ```

3. Launch an EC2 instance into the same VPC in which your domain (`d-1234567890`) is located. You must assign an IAM role to your instance. You must also ensure that your instance has a public IP address, unless you're using a NAT instance for Internet communication. Take note of the instance ID in the output.

   AWS CLI

   ```
   aws ec2 run-instances --image-id ami-1a2b3c4d --subnet-id subnet-33cc44dd
   --key-name my-key-pair --instance-type m1.large --iam-profile MyInstancePro
   file --associate-public-ip-address

   {
       "OwnerId": "123456789101",
       "ReservationId": "r-bbaa1122",
       "Groups": [
           {
               "GroupName": "default",
               "GroupId": "sg-5c5c5c5c"
           }
       ],
       "Instances": [
     ...
           "InstanceId": "i-11aa22bb",
     ...
   }
   ```

   Tools for Windows PowerShell

   ```
   New-EC2Instance -ImageId ami-1a2b3c4d -SubnetId subnet-33cc44dd -KeyName
   my-key-pair -InstanceType m1.large -InstanceProfile_Id MyInstanceProfile
   -associatePublicIp $true
   ```

4. Associate the configuration document with the running instance.

   AWS CLI

   ```
   aws ssm create-association --instance-id i-11aa22bb --name "My_Custom_Con
   fig_File"
   ```

   Tools for Windows PowerShell

   ```
   New-SSMAssociation -InstanceId i-11aa22bb -Name "My_Custom_Config_File"
   ```

5. Check the status of the domain join. For more information, see Getting the Domain Join Status (p. 296).

Joining a Domain Using the Amazon EC2 Launch Wizard

You can use the launch instance wizard in the Amazon EC2 console to join a new instance to a domain that you specify. If you don't already have one, the wizard creates a configuration document for you, and associates it with your new instance.

> **Note**
> You cannot use the Amazon EC2 console to associate a configuration document with an existing instance.

To join a domain using the launch wizard

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the Amazon EC2 console, click **Launch Instance**.
3. On the first page of the wizard, select a Windows AMI. On the next page, select an instance type, and then click **Next: Configure Instance Details**.
4. On the **Step 3: Configure Instance Details** page, select a VPC from the **Network** list, and a subnet from the **Subnet** list. Ensure that you select the VPC in which your AWS Directory Service domain is located.
5. In the **Auto-assign Public IP** list, select **Enable** (if the subnet setting is not set to enable by default).

   > **Note**
   > If you're launching your instance into a private subnet and using a NAT instance in a public subnet for Internet communication, you do not have to assign your instance a public IP address.

6. Select your domain from the **Domain join directory** list, and select the IAM role to associate with the instance from the **IAM role** list.
7. Complete the rest of the configuration steps as required, and then click **Next** until you reach the **Step 6: Configure Security Group** page. Ensure that you select or create a security group with a rule that allows RDP access from your IP address, or from a range of IP addresses within your network. For more information about security group rules, see Authorizing Inbound Traffic for Your Windows Instances (p. 448).
8. Click **Review and Launch** to launch your instance.
9. Check the status of the domain join. For more information, see Getting the Domain Join Status (p. 296).

Getting the Domain Join Status

You can check the status of your domain join by viewing the system log for the instance, or by checking the status of the association.

> **Note**
> After a configuration file is associated with an instance, it may take several minutes before the instance is joined to the domain.

You can check your instance's system log by using the Amazon EC2 console, AWS CLI, or Tools for Windows PowerShell.

To get the system log using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Instances**.
3. Select your instance, right-click, select **Instance Settings**, and then click **Get System Log**.

To get the system log using a command line tool

- Use the get-console-output (AWS CLI) command; for example:

```
aws ec2 get-console-output --instance-id i-11aa22bb
```

- Use the Get-EC2ConsoleOutput (AWS Tools for Windows PowerShell) command; for example:

```
Get-EC2ConsoleOutput -instanceId i-11aa22bb
```

In the system log, the following output indicates that the domain join was successful:

```
2015/02/02 10:59:36Z: Info: EC2Config configuration status:2;region:us-east-
1;iam:1;authz:1
2015/02/02 10:59:42Z: Info: EC2Config: Downloading config awsconfig_Domain_d-
1234567890_corp.example.com
2015/02/02 10:59:45Z: Info: EC2Config: The instance is joining domain with id:d-
1234567890, name:corp.example.com ...
2015/02/02 10:59:48Z: Info: EC2Config: The instance successfully joined the
domain.
2015/02/02 10:59:48Z: Info: EC2Config: The instance will reboot shortly for
domain join to take effect.
```

Alternatively, you can check the status of the association between the configuration document and the instance by using the AWS CLI or the Tools for Windows PowerShell.

To check the status of the association

- Use the describe-association (AWS CLI) command; for example:

```
aws ssm describe-association --name "My_Custom_Config_File" --instance-id i-
11aa22bb
```

- Use the Get-SSMAssociation (Tools for Windows PowerShell) command; for example:

```
Get-SSMAssociation -Name "My_Custom_Config_File" -instanceId i-11aa22bb
```

Connecting To Your Instance Using Domain Credentials

After you've joined your instance to a domain, you can connect to your instance using domain credentials that you've defined in AWS Directory Service.

To connect to an instance as an administrator using your directory credentials

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Instances**, select your instance, and then click **Connect**.
3. In the dialog box, click **Download Remote Desktop File**, and open the file using an RDP client.
4. On the login screen, instead of using the local computer name and password generated from your key pair file, enter the details as follows:

   - **User name**: enter the fully-qualified name of your domain, followed by a backslash (\), and then the user name, in this case, `Administrator`; for example: `corp.example.com\Administrator`.
   - **Password**: enter the password that you specified when you created your domain.

   For more information about connecting to an instance, see Connecting to Your Windows Instance Using RDP (p. 216).

After you've verified that you can connect to your instance as an administrator, users in your domain can connect to the instance using the same procedure, replacing the `Administrator` credentials with their own user name and password.

Troubleshooting

If you are having trouble joining your instance to a domain, or if you are having trouble connecting to your instance using domain credentials, first verify the status of the domain join by checking instance's system log, or by checking the status of the association: Getting the Domain Join Status (p. 296).

Cannot Connect to Instance

If the domain join was successful, but you are having trouble logging into to your instance, try the following:

- If you can connect to your instance, but you cannot log in, check that you are using the correct user name and password. The user name must include the fully qualified name of your domain (for example, `corp.example.com`), and the password must be the password configured in the domain, not the password generated by a key pair file.
- If you cannot connect to your instance, check your security group settings. You must have a rule that allows RDP access from your IP address or network.

The Domain Join was Unsuccessful

In the system log, the following output indicates the EC2Config service was unable to connect and download the associated configuration document, and therefore the domain join was unsuccessful:

```
Info: EC2Config configuration status:3;region:us-east-1;iam:0;authz:0
```

The output can help you troubleshoot the cause of the failure:

- `configuration status:3`: The calls to SSM failed. Ensure that you have granted the required IAM permissions to IAM users. SSM also requires an Internet connection from your instance - your instance must have a public IP address, and must be launched into a public subnet. For more information about public subnets, see Your VPC With Subnets in the Amazon VPC User Guide.
- `iam:0`: The instance was not launched with an IAM role. You cannot join your instance to a domain if there is no IAM role associated with the instance.
- `authz:0`: The instance is not authorized to access SSM. This happens if you launched the instance without an IAM role, or if the role associated with your instance does not have the necessary permissions to access the service.

You can also troubleshoot specific reasons for a domain join failure by checking the status of the association using the describe-association (AWS CLI) command or the Get-SSMAssociation (Tools for Windows PowerShell) command. For example, the following output indicates that the IAM role associated with the instance does not have permission to use the `ds:CreateComputer` action:

```
Name                  : My_Config_Doc
InstanceId            : i-11aabb33
Date                  : 2/10/2015 1:31:45 AM
Status.Name           : Failed
Status.Date           : 2/10/2015 1:38:38 AM
Status.Message       : RunId=631148a7-894f-4684-8718-ee4cexample, status:Failed,
 code:0,
                        message:RuntimeStatusCounts=[Failed=1],
RuntimeStatus=[aws:domainJoin={Failed,User:
                      arn:aws:sts::123456789101:assumed-role/NoDomainJoinPer
mission/i-11aabb33 is not authorized to
                        perform: ds:CreateComputer}]
Status.AdditionalInfo : {agent=EC2Config,ver=x.x.xx,osver=6.2.9200,os=Windows
Server 2012 Standard,lang=en-US}
```

Viewing Information About Your Associations

You can use the AWS CLI or the AWS Tools for Windows PowerShell to view information about your associations and your configuration documents.

Action	AWS CLI	AWS Tools for Windows PowerShell
To view information about an association for a specific instance and configuration document. You can also use this command to view the status of an association.	describe-association	Get-SSMAssociation
To view information about a specified configuration document. You can also use this command to view the status of a configuration document, for example, `creating`.	describe-document	Get-SSMDocumentDescription

Action	AWS CLI	AWS Tools for Windows PowerShell
To view the contents of a specified configuration document.	get-document	Get-SSMDocument
To view a list of associations for a specified configuration document or a specified instance.	list-associations	Get-SSMAssociationList
To view a list of your configuration documents.	list-documents	Get-SSMDocumentList

Changing an Association

You can't update a configuration document after you create it. If you want to join your instance to a new domain, you must first delete the association, and then create a new association using a new configuration documentation. It can take up to 15 minutes for the configuration changes to take effect.

For more information about deleting an association, see Disassociate the Configuration Document from the Instance (p. 289). For more information about associating a new document with an instance, see Associate the Configuration Document with the Instance (p. 288).

Deleting an association does not change the configuration on the instance. Your instance is still joined to a domain until you manually remove it from the domain by modifying the network connection configuration information and system properties of the instance.

Deleting a Configuration Document

If you no longer require a configuration document, you can delete it. You must first disassociate the file from any instances it is associated with before you delete it. For more information about deleting a configuration document, see Delete the Configuration Document (p. 289).

Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs

You can use Amazon EC2 Simple Systems Manager (SSM) to configure integration with Amazon CloudWatch and Amazon CloudWatch Logs on multiple instances to monitor their log files. You can send Windows Server messages in the application, system, security, and Event Tracing (Windows) logs to Amazon CloudWatch Logs. When you enable logging for the first time, SSM sends all logs generated within 1 minute from the time that you start uploading logs for the application, system, security, and ETW logs. Logs that occurred before this time are not included. If you disable logging and then later re-enable logging, SSM sends logs from where it left off. For any custom log files and Internet Information Services (IIS) logs, SSM reads the log files from the beginning. In addition, SSM can also send performance counter data to CloudWatch.

SSM enables you to manage the configuration of your Windows instances while they are running. You create a *configuration document*, which describes configuration tasks (for example, sending performance counters to CloudWatch and logs to CloudWatch Logs), and then associate the configuration document with one or more running Windows instances. The configuration agent on the instance processes the configuration document and configures the instance as specified.

If you previously enabled CloudWatch integration in EC2Config, the SSM settings override any settings stored locally on the instance in the **C:\Program Files\Amazon\Ec2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json** file. For more information about using EC2Config to manage performance counters and logs on single instance, see Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs (p. 245).

To manage the configuration of your Windows instances using SSM, complete the following tasks.

Tasks

- Step 1: Prepare Your Instance (p. 300)
- Step 2: Create a JSON File (p. 301)
- Step 3: Configure the Region and Namespace for CloudWatch and CloudWatch Logs (p. 304)
- Step 4: Configure the Performance Counters and Logs to Send to CloudWatch and CloudWatch Logs (p. 305)
- Step 5: Configure the Flow Control (p. 311)
- Step 6: Create a Configuration Document (p. 311)
- Step 7: Associate the Configuration Document with the Instance (p. 312)

Step 1: Prepare Your Instance

To configure an instance using SSM, you must launch it with an IAM role that grants permission to use the SSM API, and ensure the instance has the latest version of the EC2Config service.

Granting Permission to use SSM and CloudWatch Logs

If you launched your Windows instance using an IAM role, you can add a policy that grants permission to use the SSM API. Otherwise, you must launch a new instance with an IAM role. For information about creating an IAM role, see Creating a Role to Delegate Permissions to an AWS Service in the *IAM User Guide*.

The following is an example of a policy document that grants an instance the permission to use SSM and send log data to CloudWatch Logs on your behalf:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessToSSM",
      "Effect": "Allow",
      "Action": [
        "ssm:DescribeAssociation",
        "ssm:ListAssociations",
        "ssm:GetDocument",
        "ssm:UpdateAssociationStatus",
        "ds:CreateComputer",
        "cloudwatch:PutMetricData",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "*"
      ]
```

```
    }
  ]
}
```

When you create an IAM role, you must associate a trust policy with the role to ensure that Amazon EC2 is allowed to assume the role.

To edit a trust policy

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**, and then choose the name of your role.
3. Under **Trust Relationships**, choose **Edit Trust Relationship**.
4. Edit the trust policy as needed, and then choose **Update Trust Policy**. The following is an example of a trust policy that allows Amazon EC2 to assume the role.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {"Service": "ec2.amazonaws.com"},
    "Action": "sts:AssumeRole"
  }
}
```

Installing the Latest EC2Config

Using SSM requires EC2Config version 3.0 or later. If you launched your instance from a current Windows AMI, then it has the latest version of EC2Config. Otherwise, you can install the latest version of EC2Config on your instance. For more information, see Installing the Latest Version of EC2Config (p. 256).

If you launch Windows instances using your own AMI, you should create a new AMI after you install EC2Config on your instance. Consider installing other updates on your instance before you create your AMI. For more information, see Updating Your Windows Instance (p. 79) and Creating an Amazon EBS-Backed Windows AMI (p. 68).

Step 2: Create a JSON File

If you don't already have a JSON file, you must create one. Open a text editor, add the JSON to describe the configuration, and then save the file with a `.json` file extension.

For more information about the structure of the JSON for a configuration document, see Configuration Document in the *Amazon EC2 Simple Systems Manager API Reference*.

When using SSM you can only have one JSON file associated with your instance. Whether you create a new JSON file or you already have one associated with your instance, you'll need to add the following sections to it.

```
{
   "schemaVersion":"1.0",
   "description":"Example CloudWatch Logs tasks",
   "runtimeConfig":{
      "aws:cloudWatch":{
         "properties":{
```

```
            "EngineConfiguration":{
               "PollInterval":"00:00:15",
               "Components":[
                  {
                     "Id":"ApplicationEventLog",
                    "FullName":"AWS.EC2.Windows.CloudWatch.EventLog.EventLogIn
putComponent,AWS.EC2.Windows.CloudWatch",
                     "Parameters":{
                        "LogName":"Application",
                        "Levels":"value"
                     }
                  },
                  {
                     "Id":"SystemEventLog",
                    "FullName":"AWS.EC2.Windows.CloudWatch.EventLog.EventLogIn
putComponent,AWS.EC2.Windows.CloudWatch",
                     "Parameters":{
                        "LogName":"System",
                        "Levels":"value"
                     }
                  },
                  {
                     "Id":"SecurityEventLog",
                    "FullName":"AWS.EC2.Windows.CloudWatch.EventLog.EventLogIn
putComponent,AWS.EC2.Windows.CloudWatch",
                     "Parameters":{
                        "LogName":"Security",
                        "Levels":"value"
                     }
                  },
                  {
                     "Id":"ETW",
                    "FullName":"AWS.EC2.Windows.CloudWatch.EventLog.EventLogIn
putComponent,AWS.EC2.Windows.CloudWatch",
                     "Parameters":{
                        "LogName":"Microsoft-Windows-WinINet/Analytic",
                        "Levels":"value"
                     }
                  },
                  {
                     "Id":"IISLogs",
                   "FullName":"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLo
gInputComponent,AWS.EC2.Windows.CloudWatch",
                     "Parameters":{
                        "LogDirectoryPath":"path",
                        "TimestampFormat":"value",
                        "Encoding":"value",
                        "Filter":"value",
                        "CultureName":"locale",
                        "TimeZoneKind":"value",
                        "LineCount":"value"
                     }
                  },
                  {
                     "Id":"CustomLogs",
                   "FullName":"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLo
gInputComponent,AWS.EC2.Windows.CloudWatch",
                     "Parameters":{
```

```
                        "LogDirectoryPath":"path",
                        "TimestampFormat":"value",
                        "Encoding":"value",
                        "Filter":"value",
                        "CultureName":"locale",
                        "TimeZoneKind":"value",
                        "LineCount":"value"
                     }
                  },
                  {
                     "Id":"PerformanceCounter",
                    "FullName":"AWS.EC2.Windows.CloudWatch.PerformanceCounter
Component.PerformanceCounterInputComponent,AWS.EC2.Windows.CloudWatch",
                     "Parameters":{
                        "CategoryName":"name",
                        "CounterName":"name",
                        "InstanceName":"name",
                        "MetricName":"name",
                        "Unit":"unit",
                        "DimensionName":"name",
                        "DimensionValue":"value"
                     }
                  },
                  {
                     "Id":"CloudWatchLogs",
                     "FullName":"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOut
put,AWS.EC2.Windows.CloudWatch",
                     "Parameters":{
                        "AccessKey":"access-key-id",
                        "SecretKey":"secret-access-key",
                        "Region":"region",
                        "LogGroup":"group",
                        "LogStream":"stream"
                     }
                  },
                  {
                     "Id":"CloudWatch",
                     "FullName":"AWS.EC2.Windows.CloudWatch.CloudWatch.Cloud
WatchOutputComponent,AWS.EC2.Windows.CloudWatch",
                     "Parameters":{
                        "AccessKey":"access-key-id",
                        "SecretKey":"secret-access-key",
                        "Region":"region",
                        "NameSpace":"namespace"
                     }
                  }
               ],
               "Flows":{
                  "Flows":[
                     "source,destination",
                     "(source1, source2),destination",
                     "source, (destination1,destination2)"
                  ]
               }
            }
         }
      }
```

```
    }
}
```

Step 3: Configure the Region and Namespace for CloudWatch and CloudWatch Logs

Next, you'll define the credentials, region, and metric namespace that comprise the destination where your data is sent.

To set the credentials, region, and metric namespace for CloudWatch

This section of the JSON file defines the credentials, region, and metric namespace that comprise the destination where your data is sent. You can add additional sections with unique IDs (for example, "CloudWatch2", CloudWatch3", etc.) and specify a different region for each new ID to send the same data to different locations.

> **Note**
> You only need to set CloudWatch credentials if you are using EC2Config and plan to send performance counters to CloudWatch. If you're using Amazon EC2 Simple Systems Manager, your credentials are configured in the IAM role you used when you launched your Amazon EC2 instance.

1. In the JSON file, locate the **CloudWatch** section.

```
{
    "Id": "CloudWatch",
    "FullName": "AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputCom
ponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "AccessKey": "",
        "SecretKey": "",
        "Region": "us-west-1",
        "NameSpace": "Windows/Default"
    }
},
```

2. In the **AccessKey** parameter, enter your access key ID. This is not supported if you launched your instance using an IAM role. For more information, see IAM Roles for Amazon EC2 (p. 442).
3. In the **SecretKey** parameter, enter your secret access key. This is not supported if you launched your instance using an IAM role. For more information, see IAM Roles for Amazon EC2 (p. 442).
4. In the **Region** parameter, enter the region where you want to send log data. You can specify us-east-1, us-west-1, us-west-2, eu-west-1, eu-central-1, ap-southeast-1, ap-southeast-2, or ap-northeast-1. Although you can send performance counters to a different region from where you send your log data, we recommend that you set this parameter to the same region where your instance is running.
5. In the **NameSpace** parameter, enter the metric namespace where you want performance counter data to be written in CloudWatch.

To set the credentials, region, log group, and log stream for CloudWatch Logs

This section of the JSON file defines the credentials, region, log group name and log stream namespace that comprise the destination where your data is sent. You can add additional sections with unique IDs (for example, "CloudWatchLogs2", CloudWatchLogs3", etc.) and specify a different region for each new ID to send the same data to different locations.

1. In the JSON file, locate the **CloudWatchLogs** section.

```
{
    "Id": "CloudWatchLogs",
   "FullName": "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Win
dows.CloudWatch",
    "Parameters": {
        "AccessKey": "",
        "SecretKey": "",
        "Region": "us-east-1",
        "LogGroup": "Default-Log-Group",
        "LogStream": "{instance_id}"
    }
},
```

2. In the **AccessKey** parameter, enter your access key ID. This is not supported if you launched your instance using an IAM role. For more information, see IAM Roles for Amazon EC2 (p. 442).
3. In the **SecretKey** parameter, enter your secret access key. This is not supported if you launched your instance using an IAM role. For more information, see IAM Roles for Amazon EC2 (p. 442).
4. In the **Region** parameter, enter the region where you want EC2Config to send log data. You can specify us-east-1, us-west-1, us-west-2, eu-west-1, eu-central-1, ap-southeast-1, ap-southeast-2, or ap-northeast-1.
5. In the **LogGroup** parameter, enter the name for your log group. This is the same name that will be displayed on the **Log Groups** screen in the CloudWatch console.
6. In the **LogStream** parameter, enter the destination log stream. If you use **{instance_id}**, the default, EC2Config uses the instance ID of this instance as the log stream name.

   If you enter a log stream name that doesn't already exist, CloudWatch Logs automatically creates it for you. You can use a literal string or predefined variables (**{instance_id}**, **{hostname}**, **{ip_address}**, or a combination of all three to define a log stream name.

   The log stream name specified in this parameter appears on the **Log Groups > Streams for *<YourLogStream>*** screen in the CloudWatch console.

Step 4: Configure the Performance Counters and Logs to Send to CloudWatch and CloudWatch Logs

Next, you'll configure the performance counters and logs that you want to send to CloudWatch and CloudWatch Logs.

To configure the performance counters to send to CloudWatch

You can select any performance counters that are available in Performance Monitor. You can select different categories to upload to CloudWatch as metrics, such as .NET CLR Data, ASP.NET Applications, HTTP Service, Memory, or Process and Processors.

For each performance counter that you want to upload to CloudWatch, copy the **PerformanceCounter** section and change the **Id** parameter to make it unique (e.g., "PerformanceCounter2") and update the other parameters as necessary.

1. In the JSON file, locate the **PerformanceCounter** section.

```
{
    "Id": "PerformanceCounter",
   "FullName": "AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.Per
formanceCounterInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "CategoryName": "Memory",
        "CounterName": "Available MBytes",
        "InstanceName": "",
        "MetricName": "AvailableMemory",
        "Unit": "Megabytes",
        "DimensionName": "",
        "DimensionValue": ""
    }
},
```

2. In the **CategoryName** parameter, enter the performance counter category.

   a. To find the available categories and counters, open Performance Monitor.
   b. Click **Monitoring Tools**, and then click **Performance Monitor**.
   c. In the results pane, click the green **+** (plus) button.

   The categories and counters are listed in the **Add Counters** dialog box.

3. In the **CounterName** parameter, enter the name of the performance counter.
4. In the **InstanceName** parameter, in the **Add Counters** dialog box in Performance Monitor, enter one of the **Instances of selected object**. Do not use an asterisk (*) to indicate all instances because each performance counter component only supports one metric. You can, however use **_Total**.
5. In the **MetricName** parameter, enter the CloudWatch metric that you want performance data to appear under.
6. In the **Unit** parameter, enter the appropriate unit of measure for the metric:

   Seconds | Microseconds | Milliseconds | Bytes | Kilobytes | Megabytes | Gigabytes | Terabytes | Bits | Kilobits | Megabits | Gigabits | Terabits | Percent | Count | Bytes/Second | Kilobytes/Second | Megabytes/Second | Gigabytes/Second | Terabytes/Second | Bits/Second | Kilobits/Second | Megabits/Second | Gigabits/Second | Terabits/Second | Count/Second | None.
7. (optional) You can enter a dimension name and value in the **DimensionName** and **DimensionValue** parameters to specify a dimension for your metric. These parameters provide another view when listing metrics. You can also use the same dimension for multiple metrics so that you can view all metrics belonging to a specific dimension.

To send Windows application event log data to CloudWatch Logs

1. In the JSON file, locate the **ApplicationEventLog** section.

```
{
    "Id": "ApplicationEventLog",
    "FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputCompon
ent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Application",
        "Levels": "1"
    }
},
```

2. In the **Levels** parameter, enter one of the following values:

   1 - Only error messages uploaded.

   2 - Only warning messages uploaded.

   4 - Only information messages uploaded.

   You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send security log data to CloudWatch Logs

1. In the JSON file, locate the **SecurityEventLog** section.

```
{
    "Id": "SecurityEventLog",
    "FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputCompon
ent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Security",
        "Levels": "7"
    }
},
```

2. In the **Levels** parameter, enter one of the following values:

   1 - Only error messages uploaded.

   2 - Only warning messages uploaded.

   4 - Only information messages uploaded.

   You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send system event log data to CloudWatch Logs

1. In the JSON file, locate the **SystemEventLog** section.

```
{
    "Id": "SystemEventLog",
    "FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputCompon
ent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
        "Levels": "7"
    }
},
```

2. In the **Levels** parameter, enter one of the following values:

   1 - Only error messages uploaded.

2 - Only warning messages uploaded.

4 - Only information messages uploaded.

You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send other types of event log data to CloudWatch Logs

In addition to the application, system, and security logs, you can upload other types of event logs.

1. In the JSON file, add a new section.

```
{
    "Id": "",
    "FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputCompon
ent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "",
        "Levels": "7"
    }
},
```

2. In the **Id** parameter, enter a name for the log you want to upload (e.g., WindowsBackup).
3. In the **LogName** parameter, enter the name of the log you want to upload.

   a. To find the name of the log, in Event Viewer, in the navigation pane, click **Applications and Services Logs**.
   b. In the list of logs, right-click the log you want to upload (e.g., Microsoft>Windows>Backup>Operational), and then click **Create Custom View**.
   c. In the **Create Custom View** dialog box, click the **XML** tab. The **LogName** is in the <Select Path=> tag (e.g., `Microsoft-Windows-Backup`). Copy this text into the **LogName** parameter in the **AWS.EC2.Windows.CloudWatch.json** file.

4. In the **Levels** parameter, enter one of the following values:

   1 - Only error messages uploaded.

   2 - Only warning messages uploaded.

   4 - Only information messages uploaded.

   You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send Event Tracing (Windows) data to CloudWatch Logs

ETW (Event Tracing for Windows) provides an efficient and detailed logging mechanism that applications can write logs to. Each ETW is controlled by a session manager that can start and stop the logging session. Each session has a provider and one or more consumers.

1. In the JSON file, locate the **ETW** section.

```
{
    "Id": "ETW",
    "FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputCompon
ent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Microsoft-Windows-WinINet/Analytic",
        "Levels": "7"
    }
},
```

2. In the **LogName** parameter, enter the name of the log you want to upload.

   a. To find the name of the log, in Event Viewer, on the **View** menu, click **Show Analytic and Debug Logs**.
   b. In the navigation pane, click **Applications and Services Logs**.
   c. In the list of ETW logs, right-click the log you want to upload, and then click **Enable Log**.
   d. Right-click the log again, and click **Create Custom View**.
   e. In the **Create Custom View** dialog box, click the **XML** tab. The **LogName** is in the <Select Path=> tag (e.g., `Microsoft-Windows-WinINet/Analytic`). Copy this text into the **LogName** parameter in the **AWS.EC2.Windows.CloudWatch.json** file.

3. In the **Levels** parameter, enter one of the following values:

   1 - Only error messages uploaded.

   2 - Only warning messages uploaded.

   4 - Only information messages uploaded.

   You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send custom logs (any text-based log file) to CloudWatch Logs

1. In the JSON file, locate the **CustomLogs** section.

```
{
    "Id": "CustomLogs",
    "FullName": "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputCompon
ent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\CustomLogs\\",
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "Local",
        "LineCount": "5"
    }
},
```

2. In the **LogDirectoryPath** parameter, enter the path where logs are stored on your instance.

3. In the **TimestampFormat** parameter, enter the timestamp format you want to use. For a list of supported values, see the Custom Date and Time Format Strings topic on MSDN.

   Note
   Your source log file must have the timestamp at the beginning of each log line.

4. In the **Encoding** parameter, enter the file encoding to use (e.g., UTF-8). For a list of supported values, see the Encoding Class topic on MSDN.

   Note
   Use the encoding name, not the display name, as the value for this parameter.

5. (optional) In the **Filter** parameter, enter the prefix of log names. Leave this parameter blank to monitor all files. For a list of supported values, see the FileSystemWatcherFilter Property topic on MSDN.
6. (optional) In the **CultureName** parameter, enter the locale where the timestamp is logged. If **CultureName** is blank, it defaults to the same locale currently used by your Windows instance. For a list of supported values, see the National Language Support (NLS) API Reference topic on MSDN.

   Note
   The **div**, **div-MV**, **hu**, and **hu-HU** values are not supported.

7. (optional) In the **TimeZoneKind** parameter, enter **Local** or **UTC**. You can set this to provide time zone information when no time zone information is included in your log's timestamp. If this parameter is left blank and if your timestamp doesn't include time zone information, CloudWatch Logs defaults to the local time zone. This parameter is ignored if your timestamp already contains time zone information.
8. (optional) In the **LineCount** parameter, enter the number of lines in the header to identify the log file. For example, IIS log files have virtually identical headers. You could enter **3**, which would read the first three lines of the log file's header to identify it. In IIS log files, the third line is the date and time stamp, which is different between log files.

To send IIS log data to CloudWatch Logs

1. In the JSON file, locate the **IISLog** section.

```
{
    "Id": "IISLogs",
    "FullName": "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputCompon
ent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "UTC",
        "LineCount": "3"
    }
},
```

2. In the **LogDirectoryPath** parameter, enter the folder where IIS logs are stored for an individual site (e.g., **C:\\inetpub\\logs\\LogFiles\\W3SVC*n***).

   Note
   Only W3C log format is supported. IIS, NCSA, and Custom formats are not supported.

3. In the **TimestampFormat** parameter, enter the timestamp format you want to use. For a list of supported values, see the Custom Date and Time Format Strings topic on MSDN.
4. In the **Encoding** parameter, enter the file encoding to use (e.g., UTF-8). For a list of supported values, see the Encoding Class topic on MSDN.

Note
Use the encoding name, not the display name, as the value for this parameter.

5. (optional) In the **Filter** parameter, enter the prefix of log names. Leave this parameter blank to monitor all files. For a list of supported values, see the FileSystemWatcherFilter Property topic on MSDN.
6. (optional) In the **CultureName** parameter, enter the locale where the timestamp is logged. If **CultureName** is blank, it defaults to the same locale currently used by your Windows instance. For a list of supported values, see the National Language Support (NLS) API Reference topic on MSDN.

   Note
   The **div**, **div-MV**, **hu**, and **hu-HU** values are not supported.
7. (optional) In the **TimeZoneKind** parameter, enter **Local** or **UTC**. You can set this to provide time zone information when no time zone information is included in your log's timestamp. If this parameter is left blank and if your timestamp doesn't include time zone information, CloudWatch Logs defaults to the local time zone. This parameter is ignored if your timestamp already contains time zone information.
8. (optional) In the **LineCount** parameter, enter the number of lines in the header to identify the log file. For example, IIS log files have virtually identical headers. You could enter **3**, which would read the first three lines of the log file's header to identify it. In IIS log files, the third line is the date and time stamp, which is different between log files.

Step 5: Configure the Flow Control

In order to send performance counter data to CloudWatch or to send log data to CloudWatch Logs, each data type must have a corresponding destination listed in the **Flows** section. For example, to send a performance counter defined in the **"Id": "PerformanceCounter"** section of the JSON file to the CloudWatch destination defined in the **"Id": "CloudWatch"** section of the JSON file, you would enter **"PerformanceCounter,CloudWatch"** in the **Flows** section. Similarly, to send the custom log, ETW log, and system log to CloudWatch Logs, you would enter **"(CustomLogs, ETW,SystemEventLog),CloudWatchLogs"**. In addition, you can send the same performance counter or log file to more than one destination. For example, to send the application log to two different destinations that you defined in the **"Id": "CloudWatchLogs"** section of the JSON file, you would enter **"ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"** in the **Flows** section.

1. In the JSON file, locate the **Flows** section.

```
"Flows": {
    "Flows": [
      "PerformanceCounter,CloudWatch",
      "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
      "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
      "CustomLogs, CloudWatchLogs2",
      "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
    ]
}
```

2. In the **Flows** parameter, enter each data type that you want to upload (e.g., ApplicationEventLog) and destination where you want to send it (e.g., CloudWatchLogs).

Step 6: Create a Configuration Document

Use the AWS CLI or the Tools for Windows PowerShell to create a configuration document, specifying the JSON file that you created in the previous task.

AWS CLI

Use the following create-document command to name this configuration and make it available for use.

```
aws ssm create-document --content file://my-config.json --name "my-custom-config"
```

Tools for Windows PowerShell

Use the following New-SSMDocument command to name this configuration and make it available for use.

```
$doc = Get-Content my-config.json | Out-String
New-SSMDocument -Content $doc -Name "my-custom-config"
```

Step 7: Associate the Configuration Document with the Instance

Use the AWS CLI or the Tools for Windows PowerShell to associate a configuration document with an instance. You'll specify the name of the configuration document that you created in the previous task. An instance can be associated with one configuration document at a time. If you associate a configuration document with an instance that already has an associated configuration document, the new configuration document replaces the existing configuration document.

AWS CLI

Use the following create-association command to associate your configuration document with your Windows instance.

```
aws ssm create-association --instance-id i-1a2b3c4d --name "my-custom-config"
```

Tools for Windows PowerShell

Use the following New-SSMAssociation command to associate your configuration document with your Windows instance.

```
New-SSMAssociation -InstanceId i-1a2b3c4d -Name "my-custom-config"
```

To stop sending logs to CloudWatch Logs, you can disassociate the configuration document from the instance. For more information, see Disassociate the Configuration Document from the Instance (p. 289).

After you disassociate the configuration document from the instance, you can delete it. For more information, see Delete the Configuration Document (p. 289).

Configuring a Secondary Private IP Address for Your Windows Instance in a VPC

In EC2-VPC, you can specify multiple private IP addresses for your instances. After you assign a secondary private IP address to an instance in a VPC, you must configure the operating system on the instance to recognize the secondary private IP address.

Configuring the operating system on a Windows instance to recognize a secondary private IP address requires the following:

- Step 1: Configure Static IP Addressing on Your Windows Instance (p. 313)
- Step 2: Configure a Secondary Private IP Address for Your Windows Instance (p. 315)
- Step 3: Configure Applications to Use the Secondary Private IP Address (p. 316)

Note
These instructions are based on Windows Server 2008 R2. The implementation of these steps may vary based on the operating system of the Windows instance.

Prerequisites

Before you begin, make sure you meet the following requirements:

- As a best practice, launch your Windows instances using the latest AMIs. If you are using an older Windows AMI, ensure that it has the Microsoft hot fix referenced in http://support.microsoft.com/kb/2582281.
- After you launch your instance in your VPC, add a secondary private IP address. For more information, see Multiple Private IP Addresses (p. 480).
- To allow Internet requests to your website after you complete the tasks in these steps, you must configure an Elastic IP address and associate it with the secondary private IP address. For more information, see Associating an Elastic IP Address with the Secondary Private IP Address (p. 483).

Step 1: Configure Static IP Addressing on Your Windows Instance

To enable your Windows instance to use multiple IP addresses, you must configure your instance to use static IP addressing rather than a DHCP server.

Important
When you configure static IP addressing on your instance, the IP address must match exactly what is shown in the AWS console, CLI, or API. If you enter these IP addresses incorrectly, the instance could become unreachable.

To configure static IP addressing on a Windows instance

1. Connect to your instance.
2. Find the IP address, subnet mask, and default gateway addresses for the instance by performing the following steps:
   a. Click **Start**. In the **Search** field, type **cmd** to open a command prompt window, and then press **Enter**.
   b. At the command prompt, run the following command: **ipconfig /all**. Review the following section in your output, and note the **IPv4 Address**, **Subnet Mask**, **Default Gateway**, and **DNS Servers** values for the network interface.

```
Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . :
   Physical Address  . . . . . . . . :
   DHCP Enabled. . . . . . . . . . . :
```

```
   Autoconfiguration Enabled . . . . :
   IPv4 Address. . . . . . . . . . . : 10.0.0.131
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.0.0.1
   DNS Servers . . . . . . . . . . . : 10.1.1.10
                                       10.1.1.20
```

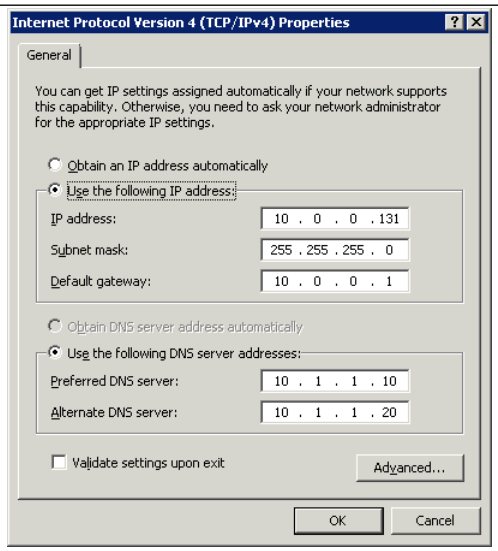
3. Open the **Network and Sharing Center** by running the following command from the command prompt:

```
C:\> %SystemRoot%\system32\control.exe ncpa.cpl
```

4. Right-click the network interface (Local Area Connection) and select **Properties**.
5. Select **Internet Protocol Version 4 (TCP/IPv4)** and click **Properties**.
6. In the **Internet Protocol Version 4 (TCP/IPv4) Properties** dialog box, select **Use the following IP address**, enter the following values, and click **OK**.

Field	Value
IP address	The IPv4 address obtained in step 2 above.
Subnet mask	The subnet mask obtained in step 2 above.
Default gateway	The default gateway address obtained in step 2 above.
Preferred DNS server	The DNS server obtained in step 2 above.
Alternate DNS server	The alternate DNS server obtained in step 2 above. If an alternate DNS server was not listed, leave this field blank.

Important
If you set the IP address to any value other than the current IP address, you will lose connectivity to the instance.



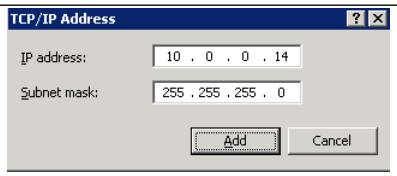
You will lose RDP connectivity to the Windows instance for a few seconds while the instance converts from using DHCP to static addressing. The instance retains the same IP address information as before, but now this information is static and not managed by DHCP.

Step 2: Configure a Secondary Private IP Address for Your Windows Instance

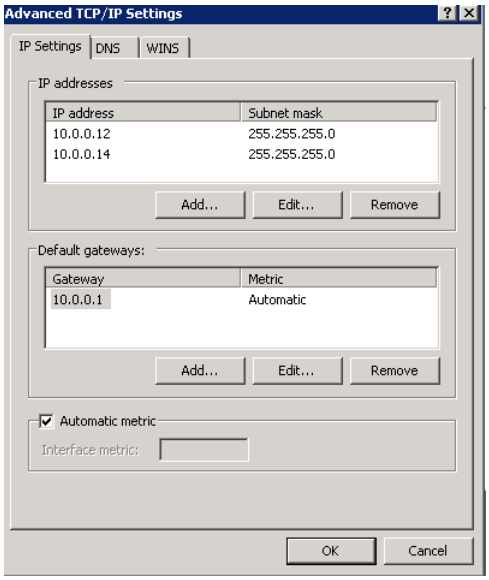
After you have set up static IP addressing on your Windows instance, you are ready to prepare a second private IP address.

To configure a secondary IP address for a Windows instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Instances**.
3. Select your instance.
4. On the **Description** tab, note the secondary IP address.
5. Connect to your instance.
6. On your Windows instance, click **Start**, and then click **Control Panel**.
7. Click **Network and Internet**, and then click **Network and Sharing Center**.
8. Click the network interface (Local Area Connection).
9. Click **Properties**.
10. In the **Local Area Connection Properties** page, click **Internet Protocol Version 4 (TCP/IPv4)**, click **Properties**, and then click **Advanced**.
11. Click **Add**.
12. In the **TCP/IP Address** dialog box, type the secondary private IP address in the **IP address** box. In the **Subnet mask** box, type the same subnet mask that you entered for the primary private IP address in Step 1: Configure Static IP Addressing on Your Windows Instance (p. 313), and then click **Add**.



13. Verify the IP address settings, and then click **OK**.



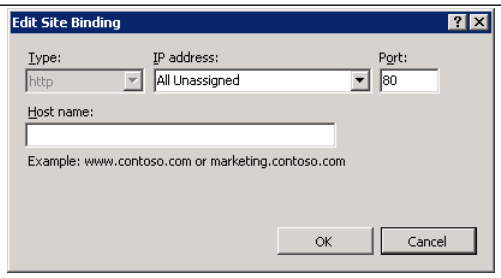
14. Click **OK** again, and then click **Close**.
15. To confirm that the secondary IP address has been added to the operating system, at a command prompt, run the command **ipconfig /all**.

Step 3: Configure Applications to Use the Secondary Private IP Address

You can configure any applications to use the secondary private IP address. For example, if your instance is running a website on IIS, you can configure IIS to use the secondary private IP address.

To configure IIS to use the secondary private IP address

1. Connect to your instance.
2. Open Internet Information Services (IIS) Manager.
3. In the **Connections** pane, expand **Sites**.
4. Right-click your website, and then click **Edit Bindings**.
5. In the **Site Bindings** dialog box, under **Type**, click **http**, and then click **Edit**.
6. In the **Edit Site Binding** dialog box, in the **IP address** box, click the secondary private IP address. (By default, each website accepts HTTP requests from all IP addresses.)



7. Click **OK**, and then click **Close**.

Monitoring Amazon EC2

Monitoring is an important part of maintaining the reliability, availability, and performance of your Amazon Elastic Compute Cloud (Amazon EC2) instances and your AWS solutions. You should collect monitoring data from all of the parts in your AWS solutions so that you can more easily debug a multi-point failure if one occurs. Before you start monitoring Amazon EC2, however, you should create a monitoring plan that should include:

- What are your goals for monitoring?
- What resources you will monitor?
- How often you will monitor these resources?
- What monitoring tools will you use?
- Who will perform the monitoring tasks?
- Who should be notified when something goes wrong?

After you have defined your monitoring goals and have created your monitoring plan, the next step is to establish a baseline for normal Amazon EC2 performance in your environment. You should measure Amazon EC2 performance at various times and under different load conditions. As you monitor Amazon EC2, you should store a history of monitoring data that you've collected. You can compare current Amazon EC2 performance to this historical data to help you to identify normal performance patterns and performance anomalies, and devise methods to address them. For example, you can monitor CPU utilization, disk I/O, and network utilization for your Amazon EC2 instances. When performance falls outside your established baseline, you might need to reconfigure or optimize the instance to reduce CPU utilization, improve disk I/O, or reduce network traffic.

To establish a baseline you should, at a minimum, monitor the following items:

Item to Monitor	Amazon EC2 Metric	Monitoring Script/CloudWatch Logs
CPU utilization	CPUUtilization (p. 334)	
Memory utilization		(Linux instances) Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances (Windows instances) Sending Performance Counters to CW; and Logs to CloudWatch Logs

Item to Monitor	Amazon EC2 Metric	Monitoring Script/CloudWatch Logs
Memory used		(Linux instances) Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances (Windows instances) Sending Performance Counters to CW; and Logs to CloudWatch Logs
Memory available		(Linux instances) Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances (Windows instances) Sending Performance Counters to CW; and Logs to CloudWatch Logs
Network utilization	NetworkIn (p. 334) NetworkOut (p. 334)	
Disk performance	DiskReadOps (p. 334) DiskWriteOps (p. 334)	
Disk Swap utilization (Linux instances only) Swap used (Linux instances only)		Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances
Page File utilization (Windows instances only) Page File used (Windows instances only) Page File available (Windows instances only)		Sending Performance Counters to CW; and Logs to CloudWatch Logs
Disk Reads/Writes	DiskReadBytes (p. 334) DiskWriteBytes (p. 334)	
Disk Space utilization (Linux instances only)		Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances
Disk Space used (Linux instances only)		Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances
Disk Space available (Linux instances only)		Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances

Automated and Manual Monitoring

AWS provides various tools that you can use to monitor Amazon EC2. You can configure some of these tools to do the monitoring for you, while some of the tools require manual intervention.

Topics

- Automated Monitoring Tools (p. 320)
- Manual Monitoring Tools (p. 321)

Automated Monitoring Tools

You can use the following automated monitoring tools to watch Amazon EC2 and report back to you when something is wrong:

- **System Status Checks** - monitor the AWS systems required to use your instance to ensure they are working properly. These checks detect problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue or you can resolve it yourself (for example, by stopping and restarting or terminating and replacing an instance). Examples of problems that cause system status checks to fail include:
  - Loss of network connectivity
  - Loss of system power
  - Software issues on the physical host
  - Hardware issues on the physical host

  For more information, see Status Checks for Your Instances (p. 322).
- **Instance Status Checks** - monitor the software and network configuration of your individual instance. These checks detect problems that require your involvement to repair. When an instance status check fails, typically you will need to address the problem yourself (for example by rebooting the instance or by making modifications in your operating system). Examples of problems that may cause instance status checks to fail include:
  - Failed system status checks
  - Misconfigured networking or startup configuration
  - Exhausted memory
  - Corrupted file system
  - Incompatible kernel

  For more information, see Status Checks for Your Instances (p. 322).
- **Amazon CloudWatch Alarms** - watch a single metric over a time period you specify, and perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon Simple Notification Service (Amazon SNS) topic or Auto Scaling policy. Alarms invoke actions for sustained state changes only. CloudWatch alarms will not invoke actions simply because they are in a particular state, the state must have changed and been maintained for a specified number of periods. For more information, see Monitoring Your Instances with CloudWatch (p. 330).
- **Amazon CloudWatch Logs** - monitor, store, and access your log files from Amazon EC2 instances, AWS CloudTrail, or other sources. For more information, see Monitoring Log Files.
- **Amazon EC2 Monitoring Scripts** - Perl scripts that can monitor memory, disk, and swap file usage in your instances. For more information, see Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances.
- **AWS Management Pack for Microsoft System Center Operations Manager** - links Amazon EC2 instances and the Microsoft Windows or Linux operating systems running inside them. The AWS Management Pack is an extension to Microsoft System Center Operations Manager. It uses a designated

computer in your datacenter (called a watcher node) and the Amazon Web Services APIs to remotely discover and collect information about your AWS resources. For more information, see AWS Management Pack for Microsoft System Center (p. 644).

Manual Monitoring Tools

Another important part of monitoring Amazon EC2 involves manually monitoring those items that the monitoring scripts, status checks, and CloudWatch alarms don't cover. The Amazon EC2 and CloudWatch console dashboards provide an at-a-glance view of the state of your Amazon EC2 environment.

- Amazon EC2 Dashboard shows:
  - Service Health and Scheduled Events by region
  - Instance state
  - Status checks
  - Alarm status
  - Instance metric details (In the navigation pane click **Instances**, select an instance, and then click the **Monitoring** tab)
  - Volume metric details (In the navigation pane click **Volumes**, select a volume, and then click the **Monitoring** tab)
- Amazon CloudWatch Dashboard shows:
  - Current alarms and status
  - Graphs of alarms and resources
  - Service health status

  In addition, you can use CloudWatch to do the following:
  - Graph Amazon EC2 monitoring data to troubleshoot issues and discover trends
  - Search and browse all your AWS resource metrics
  - Create and edit alarms to be notified of problems
  - See at-a-glance overviews of your alarms and AWS resources

Best Practices for Monitoring

Use the following best practices for monitoring to help you with your Amazon EC2 monitoring tasks.

- Make monitoring a priority to head off small problems before they become big ones.
- Create and implement a monitoring plan that collects monitoring data from all of the parts in your AWS solution so that you can more easily debug a multi-point failure if one occurs. Your monitoring plan should address, at a minimum, the following questions:
  - What are your goals for monitoring?
  - What resources you will monitor?
  - How often you will monitor these resources?
  - What monitoring tools will you use?
  - Who will perform the monitoring tasks?
  - Who should be notified when something goes wrong?
- Automate monitoring tasks as much as possible.
- Check the log files on your EC2 instances.

Monitoring the Status of Your Instances

You can monitor the status of your instances by viewing status checks and scheduled events for your instances. A status check gives you the information that results from automated checks performed by Amazon EC2. These automated checks detect whether specific issues are affecting your instances. The status check information, together with the data provided by Amazon CloudWatch, gives you detailed operational visibility into each of your instances.

You can also see status on specific events scheduled for your instances. Events provide information about upcoming activities such as rebooting or retirement that are planned for your instances, along with the scheduled start and end time of each event.

Contents

- Status Checks for Your Instances (p. 322)
- Scheduled Events for Your Instances (p. 326)

Status Checks for Your Instances

With instance status monitoring, you can quickly determine whether Amazon EC2 has detected any problems that might prevent your instances from running applications. Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues. You can view the results of these status checks to identify specific and detectable problems. This data augments the information that Amazon EC2 already provides about the intended state of each instance (such as `pending`, `running`, `stopping`) as well as the utilization metrics that Amazon CloudWatch monitors (CPU utilization, network traffic, and disk activity).

Status checks are performed every minute and each returns a pass or a fail status. If all checks pass, the overall status of the instance is **OK**. If one or more checks fail, the overall status is **impaired**. Status checks are built into Amazon EC2, so they cannot be disabled or deleted. You can, however create or delete alarms that are triggered based on the result of the status checks. For example, you can create an alarm to warn you if status checks fail on a specific instance. For more information, see Creating and Editing Status Check Alarms (p. 325).

Contents

- Types of Status Checks (p. 322)
- Viewing Status Checks (p. 323)
- Reporting Instance Status (p. 324)
- Creating and Editing Status Check Alarms (p. 325)

Types of Status Checks

There are two types of status checks: system status checks and instance status checks.

System Status Checks

Monitor the AWS systems required to use your instance to ensure they are working properly. These checks detect problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue, or you can resolve it yourself (for example, by stopping and starting an instance, or by terminating and replacing an instance).

The following are examples of problems that can cause system status checks to fail:

- Loss of network connectivity

- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host

Instance Status Checks

Monitor the software and network configuration of your individual instance. These checks detect problems that require your involvement to repair. When an instance status check fails, typically you will need to address the problem yourself (for example, by rebooting the instance or by making instance configuration changes).

The following are examples of problems that can cause instance status checks to fail:

- Failed system status checks
- Incorrect networking or startup configuration
- Exhausted memory
- Corrupted file system
- Status checks that occur during instance reboot or while a Windows instance store-backed instance is being bundled report an instance status check failure until the instance becomes available again.

Viewing Status Checks

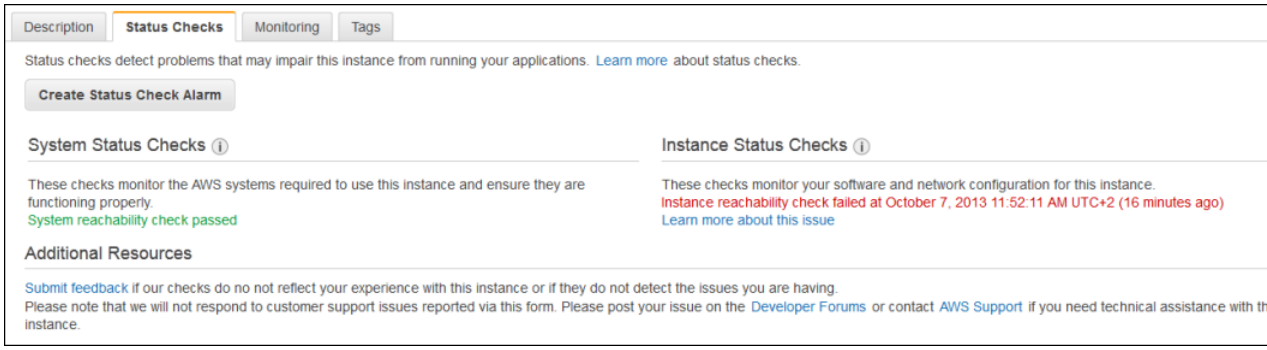
Amazon EC2 provides you with several ways to view and work with status checks.

Viewing Status Using the Console

You can view status checks using the AWS Management Console.

To view status checks using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. On the **Instances** page, the **Status Checks** column lists the operational status of each instance.
4. To view the status of a specific instance, select the instance, and then choose the **Status Checks** tab.



5. If you have an instance with a failed status check and the instance has been unreachable for over 20 minutes, choose **AWS Support** to submit a request for assistance.

Viewing Status Using the AWS CLI

You can view status checks using the describe-instance-status command.

To view the status of all instances, use the following command:

```
aws ec2 describe-instance-status
```

To get the status of all instances with a instance status of `impaired`:

```
aws ec2 describe-instance-status --filters Name=instance-status.status,Values=im
paired
```

To get the status of a single instance, use the following command:

```
aws ec2 describe-instance-status --instance-ids i-15a4417c
```

API

You can use the `DescribeInstanceStatus` action to retrieve the status of your instances. For more information, see DescribeInstanceStatus in the *Amazon EC2 API Reference*.

Reporting Instance Status

You can provide feedback if you are having problems with an instance whose status is not shown as impaired, or want to send AWS additional details about the problems you are experiencing with an impaired instance.

We use reported feedback to identify issues impacting multiple customers, but do not respond to individual account issues. Providing feedback does not change the status check results that you currently see for the instance.

Reporting Status Feedback Using the Console

To report instance status using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. Select the **Status Checks** tab, and then choose **Submit feedback**.
5. Complete the **Report Instance Status** form, and then choose **Submit**.

Reporting Status Feedback Using the AWS CLI

Use the following report-instance-status command to send feedback about the status of an impaired instance:

```
aws ec2 report-instance-status --instances i-15a4417c --status impaired --
reason-codes code
```

Reporting Status Feedback Using the API

Use the `ReportInstanceStatus` action to send feedback about the status of an instance. If your experience with the instance differs from the instance status returned by the `DescribeInstanceStatus` action, use `ReportInstanceStatus` to report your experience with the instance. Amazon EC2 collects this information to improve the accuracy of status checks. For more information, see ReportInstanceStatus in the *Amazon EC2 API Reference*.

Creating and Editing Status Check Alarms

You can create instance status and system status alarms to notify you when an instance has a failed status check.

Creating a Status Check Alarm Using the Console

You can create status check alarms for an existing instance to monitor instance status or system status. You can configure the alarm to send you a notification by email or stop, terminate, or recover an instance when it fails an instance status check or system status check.

To create a status check alarm

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. Select the **Status Checks** tab, and then choose **Create Status Check Alarm**.
5. Select **Send a notification to**. Choose an existing SNS topic, or click **create topic** to create a new one. If creating a new topic, in **With these recipients**, enter your email address and the addresses of any additional recipients, separated by commas.
6. (Optional) Choose **Take the action**, and then select the action that you'd like to take.
7. In **Whenever**, select the status check that you want to be notified about.

   Note
   If you selected **Recover this instance** in the previous step, select **Status Check Failed (System)**.
8. In **For at least**, set the number of periods you want to evaluate and in **consecutive periods**, select the evaluation period duration before triggering the alarm and sending an email.
9. (Optional) In **Name of alarm**, replace the default name with another name for the alarm.
10. Choose **Create Alarm**.

    Important
    If you added an email address to the list of recipients or created a new topic, Amazon SNS sends a subscription confirmation email message to each new address. Each recipient must confirm the subscription by clicking the link contained in that message. Alert notifications are sent only to confirmed addresses.

If you need to make changes to an instance status alarm, you can edit it.

To edit a status check alarm

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose **Actions**, select **CloudWatch Monitoring**, and then choose **Add/Edit Alarms**.
4. In the **Alarm Details** dialog box, choose the name of the alarm.

5. In the **Edit Alarm** dialog box, make the desired changes, and then choose **Save**.

Creating a Status Check Alarm Using the AWS CLI

In the following example, the alarm publishes a notification to an SNS topic, `arn:aws:sns:us-west-2:111122223333:my-sns-topic`, when the instance fails either the instance check or system status check for at least two consecutive periods. The metric is `StatusCheckFailed`.

To create a status check alarm using the CLI

1. Select an existing SNS topic or create a new one. For more information, see Using the AWS CLI with Amazon SNS in the *AWS Command Line Interface User Guide*.
2. Use the following list-metrics command to view the available Amazon CloudWatch metrics for Amazon EC2:

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

3. Use the following put-metric-alarm command to create the alarm:

```
aws cloudwatch put-metric-alarm --alarm-name StatusCheckFailed-Alarm-for-i-ab12345 --metric-name StatusCheckFailed --namespace AWS/EC2 --statistic Maximum --dimensions Name=InstanceId,Value=i-ab12345 --unit Count --period 300 --evaluation-periods 2 --threshold 1 --comparison-operator GreaterThanOrEqualToThreshold --alarm-actions arn:aws:sns:us-west-2:111122223333:my-sns-topic
```

Note

- `--period` is the time frame, in seconds, in which Amazon CloudWatch metrics are collected. This example uses 300, which is 60 seconds multiplied by 5 minutes.
- `--evaluation-periods` is the number of consecutive periods for which the value of the metric must be compared to the threshold. This example uses 2.
- `--alarm-actions` is the list of actions to perform when this alarm is triggered. Each action is specified as an Amazon Resource Name (ARN). This example configures the alarm to send an email using Amazon SNS.

Scheduled Events for Your Instances

AWS can schedule events for your instances, such as a reboot, stop/start, or retirement. These events do not occur frequently. If one of your instances will be affected by a scheduled event, AWS sends an email to the email address that's associated with your AWS account prior to the scheduled event, with details about the event, including the start and end date. Depending on the event, you might be able to take action to control the timing of the event.

To update the contact information for your account so that you can be sure to be notified about scheduled events, go to the Account Settings page.

Contents

- Types of Scheduled Events (p. 327)
- Viewing Scheduled Events (p. 327)
- Working with Instances Scheduled for Retirement (p. 328)

- Working with Instances Scheduled for Reboot (p. 329)
- Working with Instances Scheduled for Maintenance (p. 330)

Types of Scheduled Events

Amazon EC2 supports the following types of scheduled events for your instances:

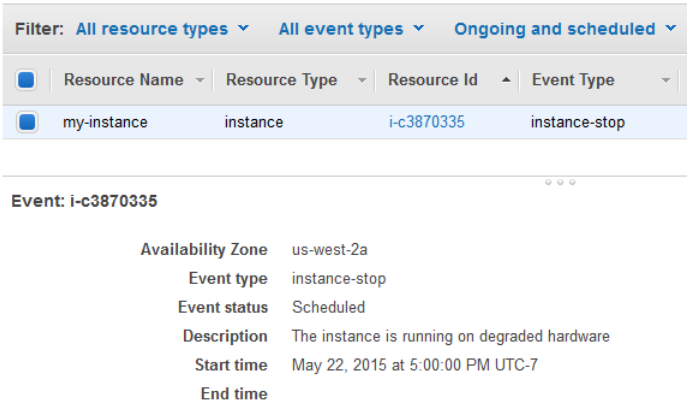
- **Instance stop**: The instance will be stopped and started to migrate it to a new host computer. Applies only to instances backed by Amazon EBS.
- **Instance retirement**: The instance will be terminated.
- **Reboot**: Either the instance will be rebooted (instance reboot) or the host computer for the instance will be rebooted (system reboot).
- **System maintenance**: The instance might be temporarily affected by network maintenance or power maintenance.

Viewing Scheduled Events

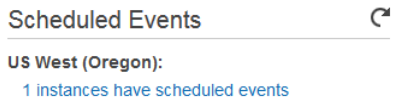
In addition to receiving notification of scheduled events in email, you can check for scheduled events.

To view scheduled events for your instances using the console

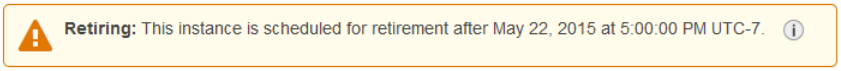
1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Events**. Any resources with an associated event are displayed. You can filter by resource type, or by specific event types. You can select the resource to view details.



3. Alternatively, in the navigation pane, choose **EC2 Dashboard**. Any resources with an associated event are displayed under **Scheduled Events**.



4. Note that events are also shown for affected resource. For example, in the navigation pane, choose **Instances**, and then select an instance. If the instance has an associated event, it is displayed in the lower pane.



To view scheduled events for your instances using the AWS CLI

Use the following describe-instance-status command:

```
aws ec2 describe-instance-status --instance-id i-1a2b3c4d
```

The following is example output showing an instance retirement event:

```
{
    "InstanceStatuses": [
        {
            "InstanceStatus": {
                "Status": "ok",
                "Details": [
                    {
                        "Status": "passed",
                        "Name": "reachability"
                    }
                ]
            },
            "AvailabilityZone": "us-west-2a",
            "InstanceId": "i-1a2b3c4d",
            "InstanceState": {
                "Code": 16,
                "Name": "running"
            },
            "SystemStatus": {
                "Status": "ok",
                "Details": [
                    {
                        "Status": "passed",
                        "Name": "reachability"
                    }
                ]
            },
            "Events": [
                {
                    "Code": "instance-stop",
                    "Description": "The instance is running on degraded hard
ware",
                    "NotBefore": "2015-05-23T00:00:00.000Z"
                }
            ]
        }
    ]
}
```

Working with Instances Scheduled for Retirement

When AWS detects irreparable failure of the underlying host computer for your instance, it schedules the instance to stop or terminate, depending on the type of root device for the instance. If the root device is

an EBS volume, the instance is scheduled to stop. If the root device is an instance store volume, the instance is scheduled to terminate. For more information, see Instance Retirement (p. 222).

Important
Any data stored on instance store volumes is lost when an instance is stopped or terminated. This includes instance store volumes that are attached to an instance that has an EBS volume as the root device. Be sure to save data from your instance store volumes that you will need later before the instance is stopped or terminated.

Actions for Instances Backed by Amazon EBS

You can wait for the instance to stop as scheduled. Alternatively, you can stop and start the instance yourself, which migrates it to a new host computer. For more information about stopping your instance, as well as information about the changes to your instance configuration when it's stopped, see Stop and Start Your Instance (p. 218).

Actions for Instances Backed by Instance Store

We recommend that you launch a replacement instance from your most recent AMI and migrate all necessary data to the replacement instance before the instance is scheduled to terminate. Then, you can terminate the original instance, or wait for it to terminate as scheduled.

Working with Instances Scheduled for Reboot

When AWS needs to perform tasks such as installing updates or maintaining the underlying host computer, it can schedule an instance or the underlying host computer for the instance for a reboot. You can determine whether the reboot event is an instance reboot or a system reboot.

To view the type of scheduled reboot event using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Events**.
3. Select **Instance resources** from the filter list, and then select your instance.
4. In the bottom pane, locate **Event type**. The value is either `system-reboot` or `instance-reboot`.

To view the type of scheduled reboot event using the AWS CLI

Use the following describe-instance-status command:

```
aws ec2 describe-instance-status --instance-ids i-15a4417c
```

Actions for Instance Reboot

You can wait for the reboot to occur within its scheduled maintenance window. Alternatively, you can reboot your instance yourself at a time that is convenient for you. For more information, see Reboot Your Instance (p. 222).

After you reboot your instance, the scheduled event for the instance reboot is canceled immediately and the event's description is updated. The pending maintenance to the underlying host computer is completed, and you can begin using your instance again after it has fully booted.

Actions for System Reboot

No action is required on your part; the system reboot occurs during its scheduled maintenance window. A system reboot typically completes in a matter of minutes. To verify that the reboot has occurred, check that there is no longer a scheduled event for the instance. We recommend that you check whether the software on your instance is operating as you expect.

Working with Instances Scheduled for Maintenance

When AWS needs to maintain the underlying host computer for an instance, it schedules the instance for maintenance. There are two types of maintenance events: network maintenance and power maintenance.

During network maintenance, scheduled instances lose network connectivity for a brief period of time. Normal network connectivity to your instance will be restored after maintenance is complete.

During power maintenance, scheduled instances are taken offline for a brief period, and then rebooted. When a reboot is performed, all of your instance's configuration settings are retained.

After your instance has rebooted (this normally takes a few minutes), verify that your application is working as expected. At this point, your instance should no longer have a scheduled event associated with it, or the description of the scheduled event begins with **[Completed]**. It sometimes takes up to 1 hour for this instance status to refresh. Completed maintenance events are displayed on the Amazon EC2 console dashboard for up to a week.

Actions for Instances Backed by Amazon EBS

You can wait for the maintenance to occur as scheduled. Alternatively, you can stop and start the instance, which migrates it to a new host computer. For more information about stopping your instance, as well as information about the changes to your instance configuration when it's stopped, see Stop and Start Your Instance (p. 218).

Actions for Instances Backed by Instance Store

You can wait for the maintenance to occur as scheduled. Alternatively, if you want to maintain normal operation during a scheduled maintenance window, you can launch a replacement instance from your most recent AMI, migrate all necessary data to the replacement instance before the scheduled maintenance window, and then terminate the original instance.

Monitoring Your Instances with CloudWatch

You can monitor your Amazon EC2 instances using Amazon CloudWatch, which collects and processes raw data from Amazon EC2 into readable, near real-time metrics. These statistics are recorded for a period of two weeks, so that you can access historical information and gain a better perspective on how your web application or service is performing. By default, Amazon EC2 metric data is automatically sent to CloudWatch in 5-minute periods. You can, however, enable detailed monitoring on an Amazon EC2 instance, which sends data to CloudWatch in 1-minute periods. For more information about Amazon CloudWatch, see the Amazon CloudWatch Developer Guide.

The following table describes basic and detailed monitoring for Amazon EC2 instances.

Type	Description
Basic	Data is available automatically in 5-minute periods at no charge.

Type	Description
Detailed	Data is available in 1-minute periods at an additional cost. To get this level of data, you must specifically enable it for the instance. For the instances where you've enabled detailed monitoring, you can also get aggregated data across groups of similar instances. For information about pricing, see the Amazon CloudWatch product page.

You can get monitoring data for your Amazon EC2 instances using either the Amazon CloudWatch API or the AWS Management Console. The console displays a series of graphs based on the raw data from the Amazon CloudWatch API. Depending on your needs, you might prefer to use either the data from the API or the graphs in the console.

Contents

- Enabling or Disabling Detailed Monitoring on an Amazon EC2 Instance (p. 331)
- View Amazon EC2 Metrics (p. 334)
- Get Statistics for Metrics (p. 340)
- Graphing Metrics (p. 357)
- Create a CloudWatch Alarm (p. 361)
- Create Alarms That Stop, Terminate, Reboot, or Recover an Instance (p. 368)

Enabling or Disabling Detailed Monitoring on an Amazon EC2 Instance

This section describes how to enable or disable detailed monitoring on either a new instance (as you launch it) or on a running or stopped instance. After you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period for the instance. You can enable or disable detailed monitoring using the console or the command line interface (CLI).

AWS Management Console

To enable detailed monitoring of an existing EC2 instance

You can enable detailed monitoring of your EC2 instances, which provides data about your instance in 1-minute periods. (There is an additional charge for 1-minute monitoring.) Detailed data is then available for the instance in the AWS Management Console graphs or through the API. To get this level of data, you must specifically enable it for the instance. For the instances on which you've enabled detailed monitoring, you can also get aggregated data across groups of similar instances. An instance must be running or stopped to enable detailed monitoring.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Instances**.
3. In the list of instances, select a running or stopped instance, click **Actions**, select **CloudWatch Monitoring**, and then click **Enable Detailed Monitoring**.
4. In the **Enable Detailed Monitoring** dialog box, click **Yes, Enable**.
5. In the **Enable Detailed Monitoring** confirmation dialog box, click **Close**.

Detailed data (collected with a 1-minute period) is then available for the instance in the AWS Management Console graphs or through the API.

To enable detailed monitoring when launching an EC2 instance

When launching an instance with the AWS Management Console, select the **Monitoring** check box on the **Configure Instance Details** page of the launch wizard.

After the instance is launched, you can select the instance in the console and view its monitoring graphs on the instance's **Monitoring** tab in the lower pane.

To disable detailed monitoring of an EC2 instance

When you no longer want to monitor your instances at 1-minute intervals, you can disable detailed monitoring and use basic monitoring instead. Basic monitoring provides data in 5-minute periods at no charge.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Instances**.
3. In the list of instances, select a running or stopped instance, click **Actions**, select **CloudWatch Monitoring**, and then click **Disable Detailed Monitoring**.
4. In the **Disable Detailed Monitoring** dialog box, click **Yes, Disable**.
5. In the **Disable Detailed Monitoring** confirmation dialog box, click **Close**.

For information about launching instances, see Launch Your Instance (p. 206).

Command Line Interface

To enable detailed monitoring on an existing instance

Use the `monitor-instances` command with one or more instance IDs. For more information about using the **monitor-instances** command, see monitor-instances in the *AWS Command Line Interface Reference*.

```
C:\> aws ec2 monitor-instances --instance-ids i-570e5a28
{
    "InstanceMonitorings": [
        {
            "InstanceId": "i-570e5a28",
            "Monitoring": {
                "State": "pending"
            }
        }
    ]
}
```

Detailed data (collected with a 1-minute period) is then available for the instance in the AWS Management Console graphs or through the API.

To enable detailed monitoring when launching an instance

Use the `run-instances` command with the `--monitoring` flag. For more information about using the **run-instances** command, see run-instances in the *AWS Command Line Interface Reference*.

```
C:\> aws ec2 run-instances --image-id ami-09092360 --key-name MyKeyPair --mon
itoring Enabled=value
```

Amazon EC2 returns output similar to the following example. The status of monitoring is listed as *pending*.

```
{
    "OwnerId": "111122223333",
    "ReservationId": "r-25fad905",
    "Groups": [
        {
            "GroupName": "default",
            "GroupId": "sg-eafe1b82"
        }
    ],
    "Instances": [
        {
            "Monitoring": {
                "State": "pending"
            },
            "PublicDnsName": null,
            "Platform": "windows",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "EbsOptimized": false,
            "LaunchTime": "2014-02-24T18:02:49.000Z",
            "ProductCodes": [],
            "StateTransitionReason": null,
            "InstanceId": "i-31283b11",
            "ImageId": "ami-09092360",
            "PrivateDnsName": null,
            "KeyName": "MyKeyPair",
            "SecurityGroups": [
                {
                    "GroupName": "default",
                    "GroupId": "sg-eafe1b82"
                }
            ],
            "ClientToken": null,
            "InstanceType": "m1.small",
            "NetworkInterfaces": [],
            "Placement": {
                "Tenancy": "default",
                "GroupName": null,
                "AvailabilityZone": "us-east-1b"
            },
            "Hypervisor": "xen",
            "BlockDeviceMappings": [],
            "Architecture": "x86_64",
            "StateReason": {
                "Message": "pending",
                "Code": "pending"
            },
            "VirtualizationType": "hvm",
            "RootDeviceType": "instance-store",
            "AmiLaunchIndex": 0
```

```
        }
    ]
}
```

After the instance is running, detailed data (collected with a 1-minute period) is then available for the instance in the AWS Management Console graphs or through the API.

To disable detailed monitoring of an instance

Use the `unmonitor-instances` command with one or more instance IDs. For more information about using the **unmonitor-instances** command, see unmonitor-instances in the *AWS Command Line Interface Reference*.

```
C:\> aws ec2 unmonitor-instances --instance-ids i-570e5a28
{
    "InstanceMonitorings": [
        {
            "InstanceId": "i-570e5a28",
            "Monitoring": {
                "State": "disabling"
            }
        }
    ]
}
```

View Amazon EC2 Metrics

Only those services in AWS that you're using send metrics to Amazon CloudWatch. You can use the Amazon CloudWatch console, the `mon-list-metrics` command, or the `ListMetrics` API to view the metrics that Amazon EC2 sends to CloudWatch. If you've enabled detailed monitoring, each data point covers the instance's previous 1 minute of activity. Otherwise, each data point covers the instance's previous 5 minutes of activity.

Metric	Description
`CPUCreditUsage`	(Only valid for T2 instances) The number of CPU credits consumed during the specified period. This metric identifies the amount of time during which physical CPUs were used for processing instructions by virtual CPUs allocated to the instance. **Note** CPU Credit metrics are available at a 5 minute frequency. Units: Count
`CPUCreditBalance`	(Only valid for T2 instances) The number of CPU credits that an instance has accumulated. This metric is used to determine how long an instance can burst beyond its baseline performance level at a given rate. **Note** CPU Credit metrics are available at a 5 minute frequency. Units: Count

Metric	Description
`CPUUtilization`	The percentage of allocated EC2 compute units that are currently in use on the instance. This metric identifies the processing power required to run an application upon a selected instance. **Note** Depending on your Amazon EC2 instance type, tools in your operating system may show a lower percentage than CloudWatch when the instance is not allocated a full processor core. Units: Percent
`DiskReadOps`	Completed read operations from all ephemeral disks available to the instance in a specified period of time. If your instance uses Amazon EBS volumes, see Amazon EBS Metrics (p. 533). **Note** To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period. Units: Count
`DiskWriteOps`	Completed write operations to all ephemeral disks available to the instance in a specified period of time. If your instance uses Amazon EBS volumes, see Amazon EBS Metrics (p. 533). **Note** To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period. Units: Count
`DiskReadBytes`	Bytes read from all ephemeral disks available to the instance (if your instance uses Amazon EBS, see Amazon EBS Metrics (p. 533).) This metric is used to determine the volume of the data the application reads from the hard disk of the instance. This can be used to determine the speed of the application. Units: Bytes
`DiskWriteBytes`	Bytes written to all ephemeral disks available to the instance (if your instance uses Amazon EBS, see Amazon EBS Metrics (p. 533).) This metric is used to determine the volume of the data the application writes onto the hard disk of the instance. This can be used to determine the speed of the application. Units: Bytes
`NetworkIn`	The number of bytes received on all network interfaces by the instance. This metric identifies the volume of incoming network traffic to an application on a single instance. Units: Bytes

Metric	Description
NetworkOut	The number of bytes sent out on all network interfaces by the instance. This metric identifies the volume of outgoing network traffic to an application on a single instance. Units: Bytes
StatusCheckFailed	A combination of StatusCheckFailed_Instance and StatusCheckFailed_System that reports if either of the status checks has failed. Values for this metric are either 0 (zero) or 1 (one.) A zero indicates that the status check passed. A one indicates a status check failure. **Note** Status check metrics are available at 1 minute frequency. For a newly launched instance, status check metric data will only be available after the instance has completed the initialization state. Status check metrics will become available within a few minutes of being in the running state. Units: Count
StatusCheckFailed_Instance	Reports whether the instance has passed the EC2 instance status check in the last minute. Values for this metric are either 0 (zero) or 1 (one.) A zero indicates that the status check passed. A one indicates a status check failure. **Note** Status check metrics are available at 1 minute frequency. For a newly launched instance, status check metric data will only be available after the instance has completed the initialization state. Status check metrics will become available within a few minutes of being in the running state. Units: Count
StatusCheckFailed_System	Reports whether the instance has passed the EC2 system status check in the last minute. Values for this metric are either 0 (zero) or 1 (one.) A zero indicates that the status check passed. A one indicates a status check failure. **Note** Status check metrics are available at 1 minute frequency. For a newly launched instance, status check metric data will only be available after the instance has completed the initialization state. Status check metrics will become available within a few minutes of being in the running state. Units: Count

You can use the dimensions in the following table to refine the metrics returned for your instances.

Dimension	Description
`AutoScalingGroupName`	This dimension filters the data you request for all instances in a specified capacity group. An *AutoScalingGroup* is a collection of instances you define if you're using the Auto Scaling service. This dimension is available only for EC2 metrics when the instances are in such an AutoScalingGroup. Available for instances with Detailed or Basic Monitoring enabled.
`ImageId`	This dimension filters the data you request for all instances running this EC2 Amazon Machine Image (AMI). Available for instances with Detailed Monitoring enabled.
`InstanceId`	This dimension filters the data you request for the identified instance only. This helps you pinpoint an exact instance from which to monitor data.
`InstanceType`	This dimension filters the data you request for all instances running with this specified instance type. This helps you categorize your data by the type of instance running. For example, you might compare data from an m1.small instance and an m1.large instance to determine which has the better business value for your application. Available for instances with Detailed Monitoring enabled.

For more information about using the `GetMetricStatistics` action, see GetMetricStatistics in the *Amazon CloudWatch API Reference*.

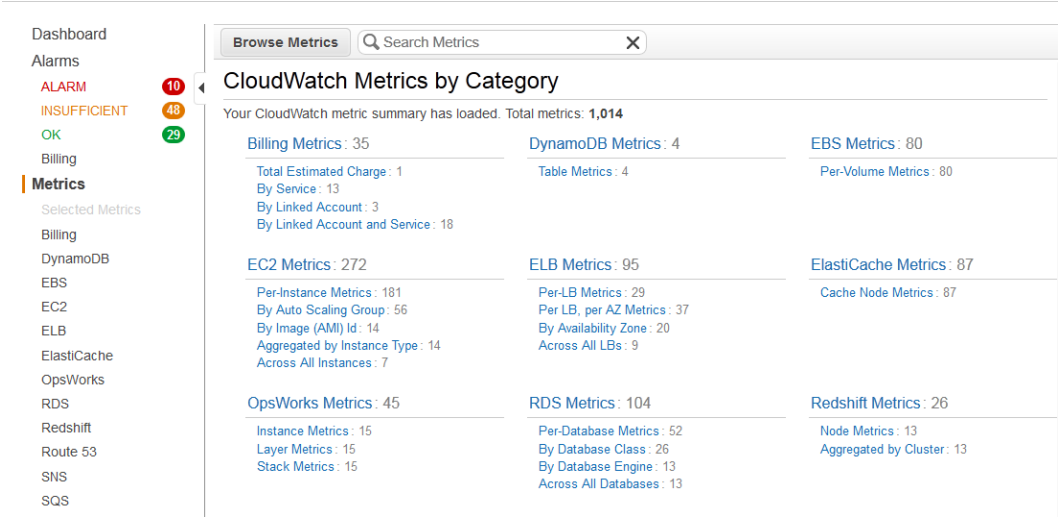
AWS Management Console

To view available metrics by category

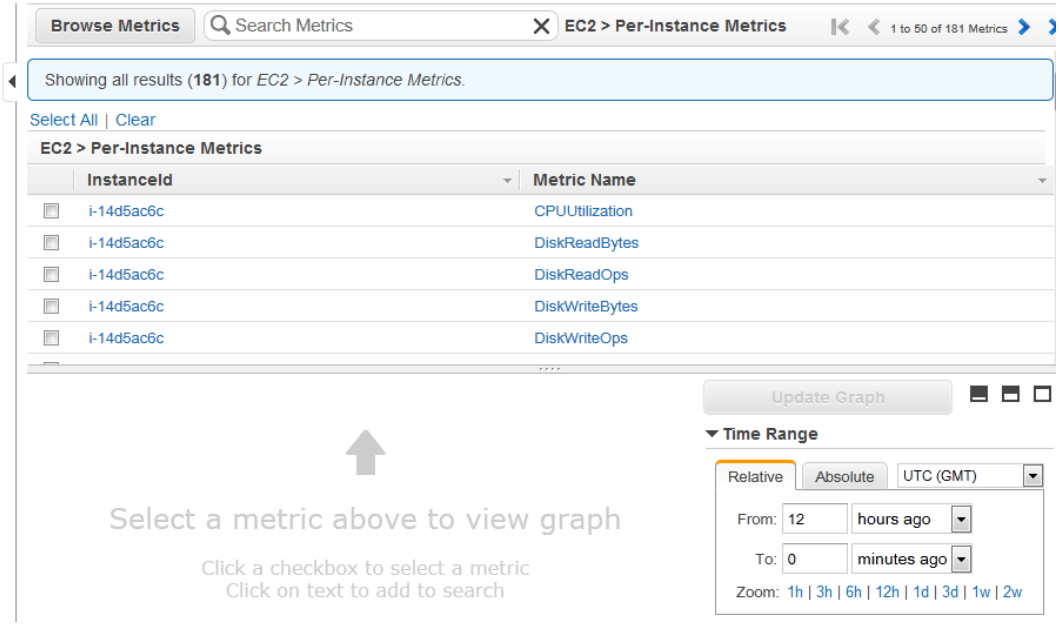
You can view metrics by category. Metrics are grouped first by Namespace, and then by the various Dimension combinations within each Namespace. For example, you can view all EC2 metrics, or EC2 metrics grouped by instance ID, instance type, image (AMI) ID, or Auto Scaling Group.

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see Regions and Endpoints.

3. In the navigation pane, click **Metrics**.



4. In the **CloudWatch Metrics by Category** pane, under **EC2 Metrics**, select **Per-Instance Metrics**, and then in the upper pane, scroll down to view the full list of metrics.



Command Line Interface

To list available metrics across multiple Amazon EC2 instances

Enter the `list-metrics` command and specify the AWS/EC2 namespace to limit the results to Amazon EC2. For more information about the `list-metrics` command, see list-metrics in the *AWS Command Line Interface Reference*.

```
C:\> aws cloudwatch list-metrics --namespace AWS/EC2
```

CloudWatch returns the following (partial listing):

```
{
            "Namespace": "AWS/EC2",
            "Dimensions": [
                {
                    "Name": "InstanceType",
                    "Value": "t1.micro"
                }
            ],
            "MetricName": "CPUUtilization"
        },
        {
            "Namespace": "AWS/EC2",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-570e5a28"
                }
            ],
            "MetricName": "DiskWriteOps"
        },
        {
            "Namespace": "AWS/EC2",
            "Dimensions": [
                {
                    "Name": "InstanceType",
                    "Value": "t1.micro"
                }
            ],
            "MetricName": "NetworkOut"
        },
        {
            "Namespace": "AWS/EC2",
            "Dimensions": [
                {
                    "Name": "ImageId",
                    "Value": "ami-6cb90605"
                }
            ],
            "MetricName": "CPUUtilization"
        },
        {
            "Namespace": "AWS/EC2",
            "Dimensions": [
                {
                    "Name": "ImageId",
                    "Value": "ami-6cb90605"
                }
            ],
            "MetricName": "NetworkIn"
        },
        {
            "Namespace": "AWS/EC2",
            "Dimensions": [
                {
                    "Name": "InstanceType",
                    "Value": "t1.micro"
```

```
                }
            ],
            "MetricName": "DiskReadBytes"
        },
        {
            "Namespace": "AWS/EC2",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-570e5a28"
                }
            ],
            "MetricName": "StatusCheckFailed_System"
        },
        {
            "Namespace": "AWS/EC2",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-570e5a28"
                }
            ],
            "MetricName": "NetworkOut"
        },
        {
            "Namespace": "AWS/EC2",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "MetricName": "DiskWriteBytes"
        }
    ]
}
```

Get Statistics for Metrics

This set of scenarios shows you how you can use the AWS Management Console, the `get-metric-statistics` command, or the `GetMetricStatistics` API to get a variety of statistics.

Note
Start and end times must be within the last 14 days.

Contents

- Get Statistics for a Specific EC2 Instance (p. 341)
- Aggregating Statistics Across Instances (p. 345)
- Get Statistics Aggregated by Auto Scaling Group (p. 349)
- Get Statistics Aggregated by Image (AMI) ID (p. 352)

Get Statistics for a Specific EC2 Instance

The following scenario walks you through how to use the AWS Management Console or the `get-metric-statistics` command to determine the maximum CPU utilization of a specific EC2 instance.

Note
Start and end times must be within the last 14 days.

For this example, we assume that you have an EC2 instance ID. You can get an active EC2 instance ID through the AWS Management Console or with the `describe-instances` command.

AWS Management Console

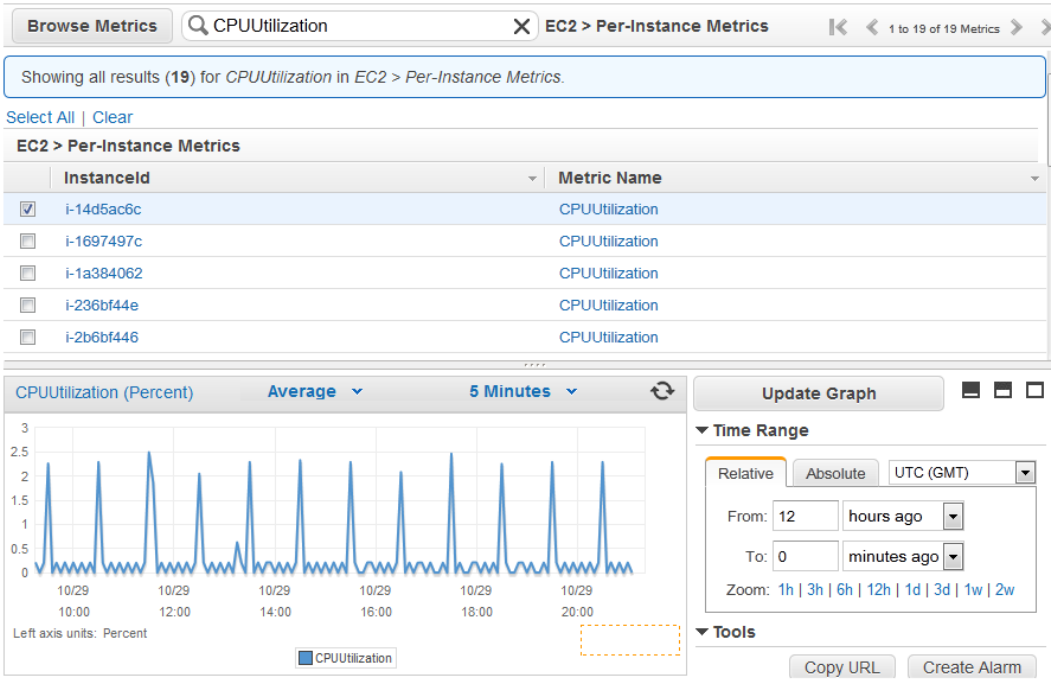
To display the average CPU utilization for a specific instance

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see Regions and Endpoints.

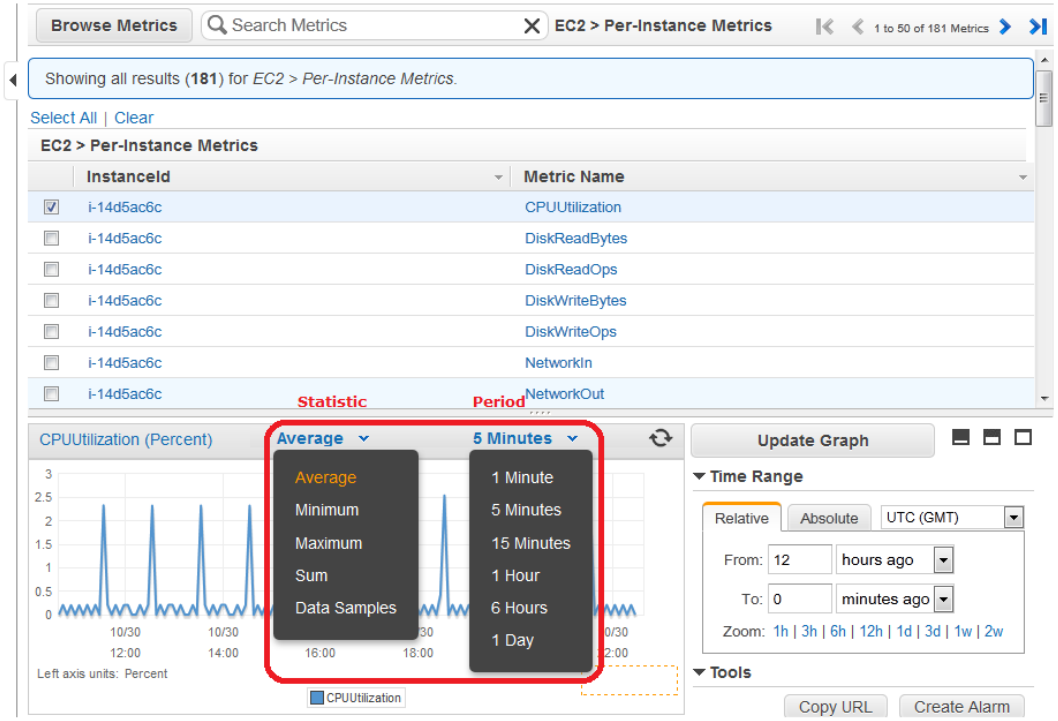
3. In the navigation pane, click **Metrics**.
4. In the **CloudWatch Metrics by Category** pane, select **EC2: Metrics**.

   The metrics available for individual instances appear in the upper pane.
5. Select a row that contains **CPUUtilization** for a specific InstanceId.

   A graph showing average `CPUUtilization` for a single instance appears in the details pane.



6. To change the **Statistic**, e.g., Average, for the metric, choose a different value from the pop-up list.



7. To change the **Period**, e.g., 5 Minutes, to view data in more granular detail, choose a different value from the pop-up list.

Command Line Interface

To get the CPU utilization per EC2 instance

Enter the `get-metric-statistics` command with the following parameters. For more information about the `get-metric-statistics` command, see get-metric-statistics in the *AWS Command Line Interface Reference*.

```
C:\> aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-time 2014-02-18T23:18:00 --end-time 2014-02-19T23:18:00 --period 3600 --namespace AWS/EC2 --statistics Maximum --dimensions Name=InstanceId,Value=<your-instance-id>
```

The AWS CLI returns the following:

```
{
    "Datapoints": [
        {
            "Timestamp": "2014-02-19T00:18:00Z",
            "Maximum": 0.33000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T03:18:00Z",
            "Maximum": 99.670000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T07:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T12:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T02:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T01:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T17:18:00Z",
            "Maximum": 3.3900000000000001,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T13:18:00Z",
            "Maximum": 0.33000000000000002,
            "Unit": "Percent"
        },
```

```
{
    "Timestamp": "2014-02-18T23:18:00Z",
    "Maximum": 0.67000000000000004,
    "Unit": "Percent"
},
{
    "Timestamp": "2014-02-19T06:18:00Z",
    "Maximum": 0.34000000000000002,
    "Unit": "Percent"
},
{
    "Timestamp": "2014-02-19T11:18:00Z",
    "Maximum": 0.34000000000000002,
    "Unit": "Percent"
},
{
    "Timestamp": "2014-02-19T10:18:00Z",
    "Maximum": 0.34000000000000002,
    "Unit": "Percent"
},
{
    "Timestamp": "2014-02-19T19:18:00Z",
    "Maximum": 8.0,
    "Unit": "Percent"
},
{
    "Timestamp": "2014-02-19T15:18:00Z",
    "Maximum": 0.34000000000000002,
    "Unit": "Percent"
},
{
    "Timestamp": "2014-02-19T14:18:00Z",
    "Maximum": 0.34000000000000002,
    "Unit": "Percent"
},
{
    "Timestamp": "2014-02-19T16:18:00Z",
    "Maximum": 0.34000000000000002,
    "Unit": "Percent"
},
{
    "Timestamp": "2014-02-19T09:18:00Z",
    "Maximum": 0.34000000000000002,
    "Unit": "Percent"
},
{
    "Timestamp": "2014-02-19T04:18:00Z",
    "Maximum": 2.0,
    "Unit": "Percent"
},
{
    "Timestamp": "2014-02-19T08:18:00Z",
    "Maximum": 0.68000000000000005,
    "Unit": "Percent"
},
{
    "Timestamp": "2014-02-19T05:18:00Z",
    "Maximum": 0.33000000000000002,
```

```
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T18:18:00Z",
            "Maximum": 6.6699999999999999,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```

The returned statistics are six-minute values for the requested two-day time interval. Each value represents the maximum CPU utilization percentage for a single EC2 instance.

Aggregating Statistics Across Instances

Aggregate statistics are available for the instances that have detailed monitoring enabled. Instances that use basic monitoring are not included in the aggregates. In addition, Amazon CloudWatch does not aggregate data across Regions. Therefore, metrics are completely separate between Regions. Before you can get statistics aggregated across instances, you must enable detailed monitoring (at an additional charge), which provides data in 1-minute periods. This scenario shows you how to use detailed monitoring with either the AWS Management Console, the `GetMetricStatistics` API, or the `get-metric-statistics` command to get the average CPU usage for your EC2 instances. Because no dimension is specified, CloudWatch returns statistics for all dimensions in the `AWS/EC2` namespace. To get statistics for other metrics, see Amazon CloudWatch Namespaces, Dimensions, and Metrics Reference.

Important
This technique for retrieving all dimensions across an AWS namespace does not work for custom namespaces that you publish to Amazon CloudWatch. With custom namespaces, you must specify the complete set of dimensions that are associated with any given data point to retrieve statistics that include the data point.

AWS Management Console

To display average CPU utilization for your Amazon EC2 instances

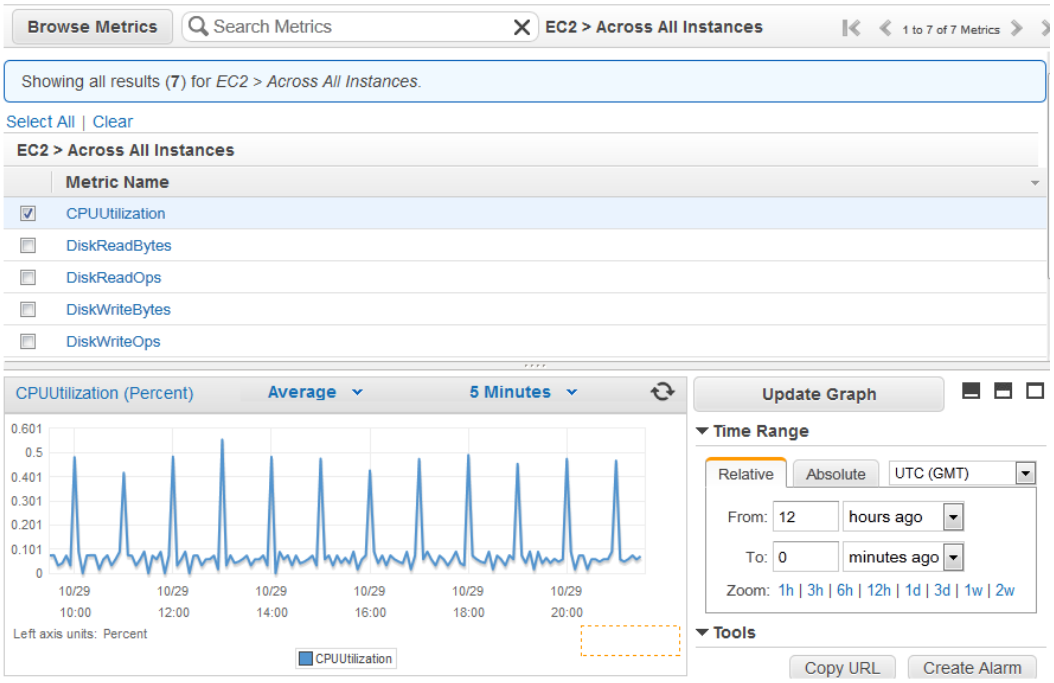
1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see Regions and Endpoints.

3. In the navigation pane, click **Metrics**.
4. In the **CloudWatch Metrics by Category** pane, under **EC2 Metrics**, select **Across All Instances**.

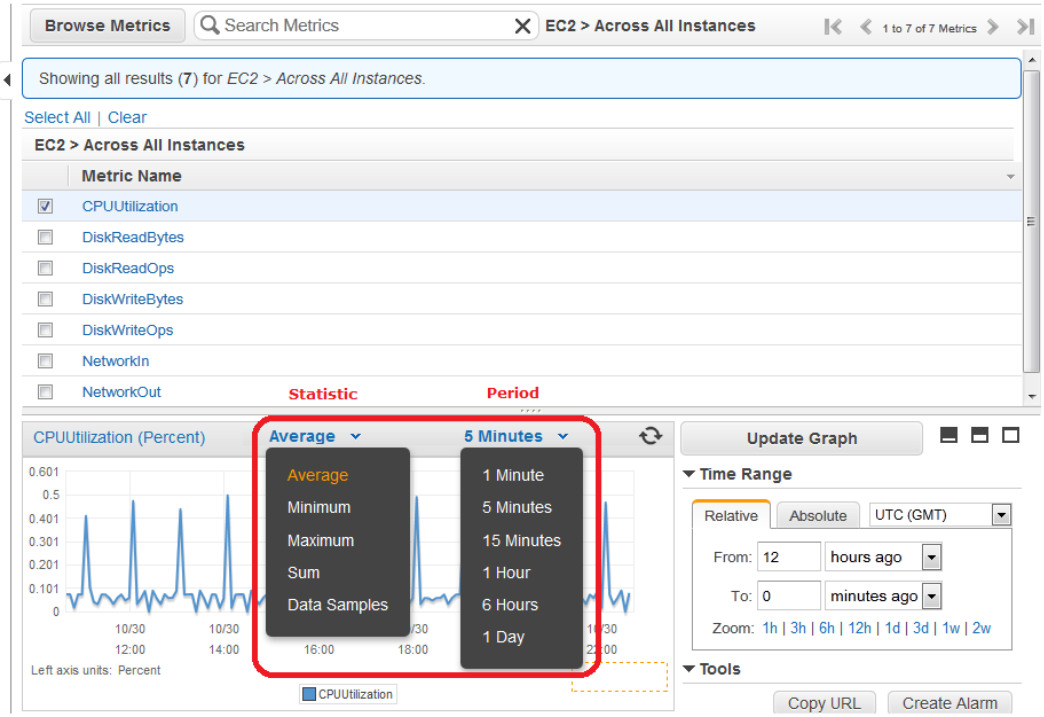
The metrics available across all instances are displayed in the upper pane.

5. In the upper pane, select the row that contains **CPUUtilization**.

   A graph showing `CPUUtilization` for your EC2 instances is displayed in the details pane.



6. To change the **Statistic**, e.g., Average, for the metric, choose a different value from the pop-up list.



7. To change the **Period**, e.g., 5 Minutes, to view data in more granular detail, choose a different value from the pop-up list.

Command Line Interface

To get average CPU utilization across your Amazon EC2 instances

Enter the `get-metric-statistics` command with the following parameters. For more information about the `get-metric-statistics` command, see get-metric-statistics in the *AWS Command Line Interface Reference*.

```
C:\> aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-time 2014-02-11T23:18:00 --end-time 2014-02-12T23:18:00 --period 3600 --namespace AWS/EC2 --statistics "Average" "SampleCount"
```

The AWS CLI returns the following:

```
{
    "Datapoints": [
        {
            "SampleCount": 238.0,
            "Timestamp": "2014-02-12T07:18:00Z",
            "Average": 0.038235294117647062,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2014-02-12T09:18:00Z",
            "Average": 0.16670833333333332,
            "Unit": "Percent"
        },
        {
            "SampleCount": 238.0,
            "Timestamp": "2014-02-11T23:18:00Z",
            "Average": 0.041596638655462197,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2014-02-12T16:18:00Z",
            "Average": 0.039458333333333345,
            "Unit": "Percent"
        },
        {
            "SampleCount": 239.0,
            "Timestamp": "2014-02-12T21:18:00Z",
            "Average": 0.041255230125523033,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2014-02-12T01:18:00Z",
            "Average": 0.044583333333333336,
            "Unit": "Percent"
        },
        {
```

```
        "SampleCount": 239.0,
        "Timestamp": "2014-02-12T18:18:00Z",
        "Average": 0.043054393305439344,
        "Unit": "Percent"
    },
    {
        "SampleCount": 240.0,
        "Timestamp": "2014-02-12T13:18:00Z",
        "Average": 0.039458333333333345,
        "Unit": "Percent"
    },
    {
        "SampleCount": 238.0,
        "Timestamp": "2014-02-12T15:18:00Z",
        "Average": 0.041260504201680689,
        "Unit": "Percent"
    },
    {
        "SampleCount": 240.0,
        "Timestamp": "2014-02-12T19:18:00Z",
        "Average": 0.037666666666666668,
        "Unit": "Percent"
    },
    {
        "SampleCount": 240.0,
        "Timestamp": "2014-02-12T06:18:00Z",
        "Average": 0.037541666666666675,
        "Unit": "Percent"
    },
    {
        "SampleCount": 240.0,
        "Timestamp": "2014-02-12T20:18:00Z",
        "Average": 0.039333333333333338,
        "Unit": "Percent"
    },
    {
        "SampleCount": 240.0,
        "Timestamp": "2014-02-12T08:18:00Z",
        "Average": 0.039250000000000014,
        "Unit": "Percent"
    },
    {
        "SampleCount": 239.0,
        "Timestamp": "2014-02-12T03:18:00Z",
        "Average": 0.037740585774058588,
        "Unit": "Percent"
    },
    {
        "SampleCount": 240.0,
        "Timestamp": "2014-02-12T11:18:00Z",
        "Average": 0.039500000000000007,
        "Unit": "Percent"
    },
    {
        "SampleCount": 238.0,
        "Timestamp": "2014-02-12T02:18:00Z",
        "Average": 0.039789915966386563,
        "Unit": "Percent"
```

```
        },
        {
            "SampleCount": 238.0,
            "Timestamp": "2014-02-12T22:18:00Z",
            "Average": 0.039705882352941181,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2014-02-12T14:18:00Z",
            "Average": 0.082458333333333328,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2014-02-12T05:18:00Z",
            "Average": 0.04287500000000001,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2014-02-12T17:18:00Z",
            "Average": 0.039458333333333345,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2014-02-12T10:18:00Z",
            "Average": 0.083416666666666667,
            "Unit": "Percent"
        },
        {
            "SampleCount": 236.0,
            "Timestamp": "2014-02-12T00:18:00Z",
            "Average": 0.036567796610169498,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2014-02-12T12:18:00Z",
            "Average": 0.039541666666666676,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2014-02-12T04:18:00Z",
            "Average": 0.043000000000000003,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```


Get Statistics Aggregated by Auto Scaling Group

Aggregate statistics are available for the instances that have detailed monitoring enabled. Instances that use basic monitoring are not included in the aggregates. In addition, Amazon CloudWatch does not

aggregate data across Regions. Therefore, metrics are completely separate between Regions. Before you can get statistics aggregated across instances, you must enable detailed monitoring (at an additional charge), which provides data in 1-minute periods.

This scenario shows you how to use the AWS Management Console, the `get-metric-statistics` command, or the `GetMetricStatistics` API with the *DiskWriteBytes* metric to retrieve the total bytes written to disk for one Auto Scaling group. The total is computed for one-minute periods for a 24-hour interval across all EC2 instances in the specified *AutoScalingGroupName*.

Note
Start and end times must be within the last 14 days.

We assume for this example that an EC2 application is running and has an Auto Scaling group named `test-group-1`.

AWS Management Console

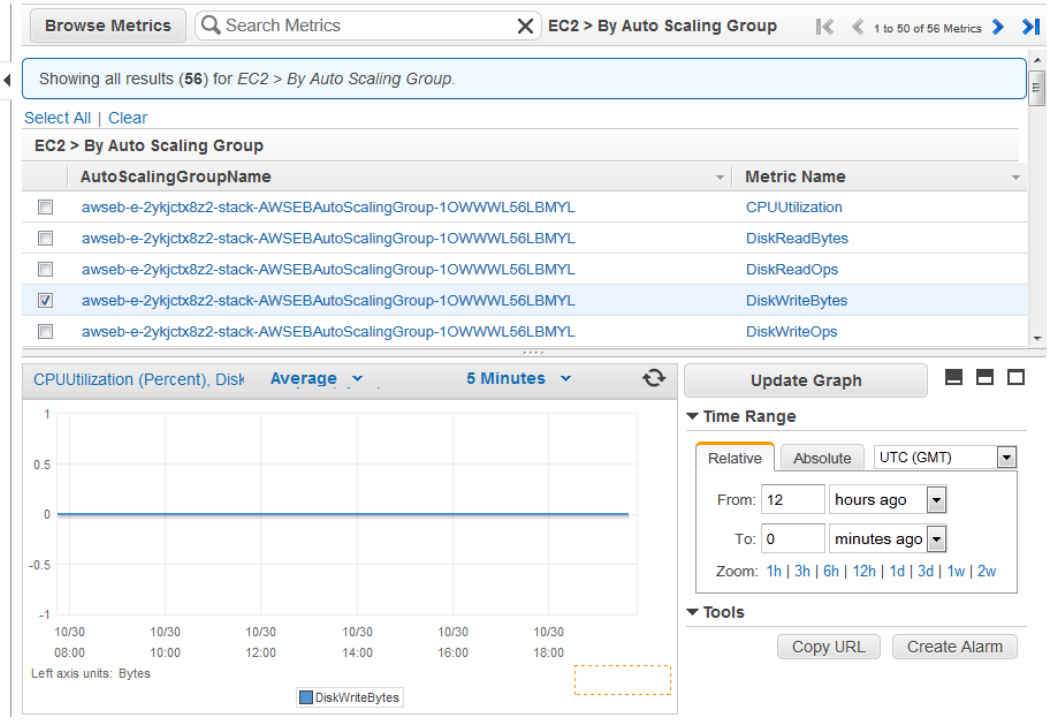
To display total DiskWriteBytes for an Auto-Scaled EC2 application

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see Regions and Endpoints.

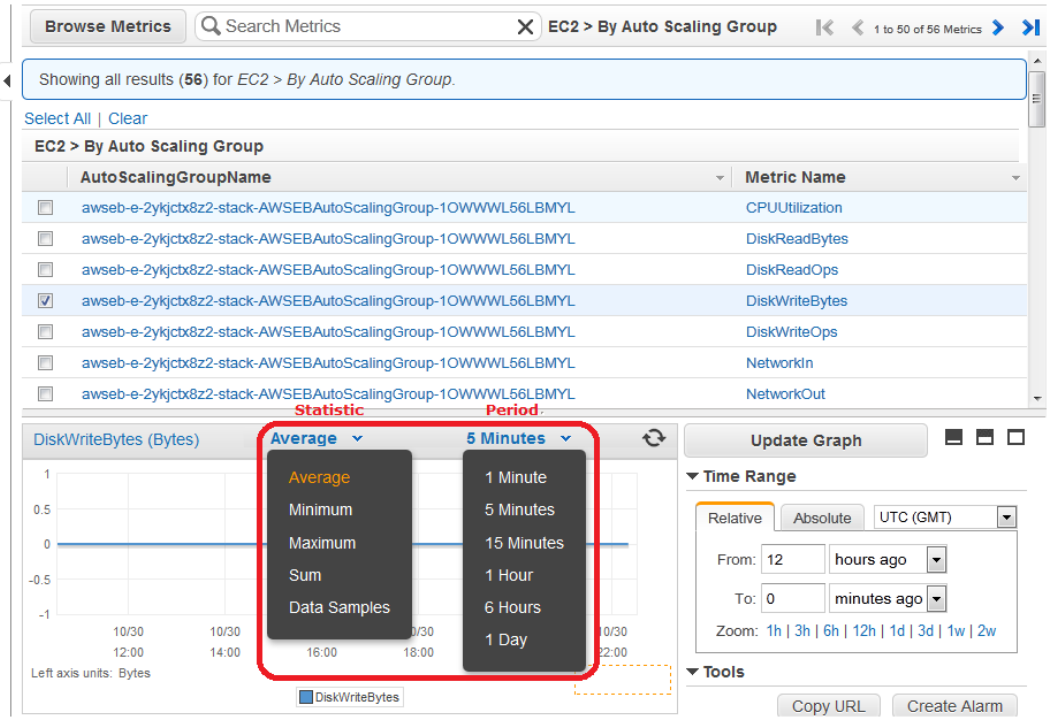
3. In the navigation pane, click **Metrics**.
4. In the **CloudWatch Metrics by Category** pane, under **EC2 Metrics**, select **By Auto Scaling Group**.

   The metrics available for Auto Scaling groups are displayed in the upper pane.
5. Select the row that contains **DiskWriteBytes**.

   A graph showing `DiskWriteBytes` for all EC2 instances appears in the details pane.



6. To change the **Statistic**, e.g., Average, for the metric, choose a different value from the pop-up list.



7. To change the **Period**, e.g., 5 Minutes, to view data in more granular detail, choose a different value from the pop-up list.

Command Line Interface

To get total DiskWriteBytes for an auto-scaled EC2 application

Enter the `get-metric-statistics` command with the following parameters. For more information about the `get-metric-statistics` command, see get-metric-statistics in the *AWS Command Line Interface Reference*.

```
C:\> aws cloudwatch get-metric-statistics --metric-name DiskWriteBytes --start-time 2014-02-16T23:18:00 --end-time 2014-02-18T23:18:00 --period 360 --namespace AWS/EC2 --statistics "Sum" "SampleCount" --dimensions Name=AutoScalingGroupName,Value=test-group-1
```

The AWS CLI returns the following:

```
{
    "Datapoints": [
        {
            "SampleCount": 18.0,
            "Timestamp": "2014-02-19T21:36:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "SampleCount": 5.0,
            "Timestamp": "2014-02-19T21:42:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        }
    ],
    "Label": "DiskWriteBytes"
}
```

Get Statistics Aggregated by Image (AMI) ID

Aggregate statistics are available for the instances that have detailed monitoring enabled. Instances that use basic monitoring are not included in the aggregates. In addition, Amazon CloudWatch does not aggregate data across Regions. Therefore, metrics are completely separate between Regions. Before you can get statistics aggregated across instances, you must enable detailed monitoring (at an additional charge), which provides data in 1-minute periods.

This scenario shows you how to use the AWS Management Console, the `get-metric-statistics` command, or the `GetMetricStatistics` API to determine average CPU utilization for all instances that match a given image ID. The average is over 60-second time intervals for a one-day period.

> **Note**
> Start and end times must be within the last 14 days.

In this scenario, the EC2 instances are running an image ID of `ami-c5e40dac`.

AWS Management Console

To display the average CPU utilization for an image ID

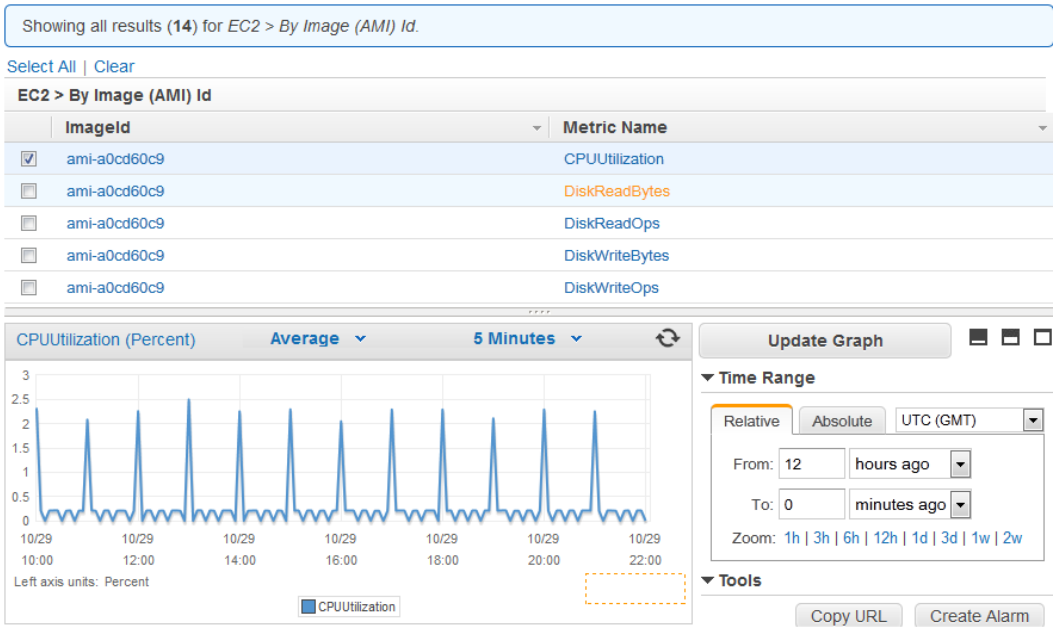
1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.

2. If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see Regions and Endpoints.

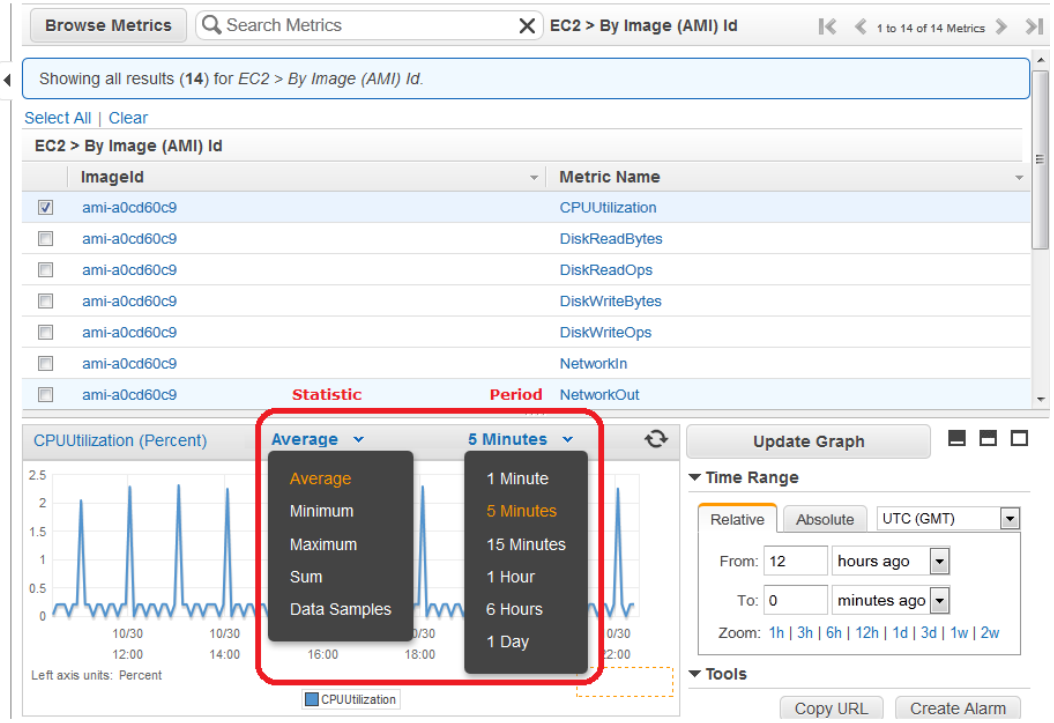
3. In the navigation pane, click **Metrics**.
4. In the **CloudWatch Metrics by Category** pane, under **EC2 Metrics**, select **By Image (AMI) Id**.

   The metrics available for image IDs appear in the upper pane.
5. Select a row that contains **CPUUtilization** and an image ID.

   A graph showing average `CPUUtilization` for all EC2 instances based on the `ami-c5e40dac` image ID appears in the details pane.



6. To change the **Statistic**, e.g., Average, for the metric, choose a different value from the pop-up list.



7. To change the **Period**, e.g., 5 Minutes, to view data in more granular detail, choose a different value from the pop-up list.

Command Line Interface

To get the average CPU utilization for an image ID

Enter the `get-metric-statistics` command as in the following example. For more information about the `get-metric-statistics` command, see get-metric-statistics in the *AWS Command Line Interface Reference*.

```
C:\> aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-time 2014-02-10T00:00:00 --end-time 2014-02-11T00:00:00 --period 3600 --statistics Average --namespace AWS/EC2 --dimensions Name="ImageId",Value=ami-3c47a355"
```

The AWS CLI returns the following:

```
{
    "Datapoints": [
        {
            "Timestamp": "2014-02-10T07:00:00Z",
            "Average": 0.041000000000000009,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T14:00:00Z",
            "Average": 0.079579831932773085,
            "Unit": "Percent"
        },
```

```
        {
            "Timestamp": "2014-02-10T06:00:00Z",
            "Average": 0.036000000000000011,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T13:00:00Z",
            "Average": 0.037625000000000013,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T18:00:00Z",
            "Average": 0.042750000000000003,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T21:00:00Z",
            "Average": 0.039705882352941188,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T20:00:00Z",
            "Average": 0.039375000000000007,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T02:00:00Z",
            "Average": 0.041041666666666671,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T01:00:00Z",
            "Average": 0.041083333333333354,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T23:00:00Z",
            "Average": 0.038016877637130804,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T15:00:00Z",
            "Average": 0.037666666666666668,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T12:00:00Z",
            "Average": 0.039291666666666676,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T03:00:00Z",
            "Average": 0.036000000000000004,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T04:00:00Z",
            "Average": 0.042666666666666672,
```

```
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T19:00:00Z",
            "Average": 0.038305084745762719,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T22:00:00Z",
            "Average": 0.039291666666666676,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T09:00:00Z",
            "Average": 0.17126050420168065,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T08:00:00Z",
            "Average": 0.041166666666666678,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T11:00:00Z",
            "Average": 0.082374999999999962,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T17:00:00Z",
            "Average": 0.037625000000000013,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T10:00:00Z",
            "Average": 0.039458333333333345,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T05:00:00Z",
            "Average": 0.039250000000000007,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T00:00:00Z",
            "Average": 0.037625000000000013,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T16:00:00Z",
            "Average": 0.041512605042016815,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```


The operation returns statistics that are one-minute values for the one-day interval. Each value represents an average CPU utilization percentage for EC2 instances running the specified machine image.

Graphing Metrics

After you launch an instance, you can go to the Amazon EC2 console and view the instance's monitoring graphs. They're displayed when you select the instance on the **Instances** page in the EC2 Dashboard. A **Monitoring** tab is displayed next to the instance's **Description** tab. The following graphs are available:

- Average CPU Utilization (Percent)
- Average Disk Reads (Bytes)
- Average Disk Writes (Bytes)
- Maximum Network In (Bytes)
- Maximum Network Out (Bytes)
- Summary Disk Read Operations (Count)
- Summary Disk Write Operations (Count)
- Summary Status (Any)
- Summary Status Instance (Count)
- Summary Status System (Count)

Each graph is based on one of the available Amazon EC2 metrics. For more information about the metrics and the data they provide to the graphs, see View Amazon EC2 Metrics (p. 334).

You can also use the CloudWatch console to graph metric data generated by Amazon EC2 and other AWS services to make it easier to see what's going on. You can use the following procedures to graph metrics in CloudWatch.

Contents

- Graph a Metric (p. 357)
- Graph a Metric Across Resources (p. 358)

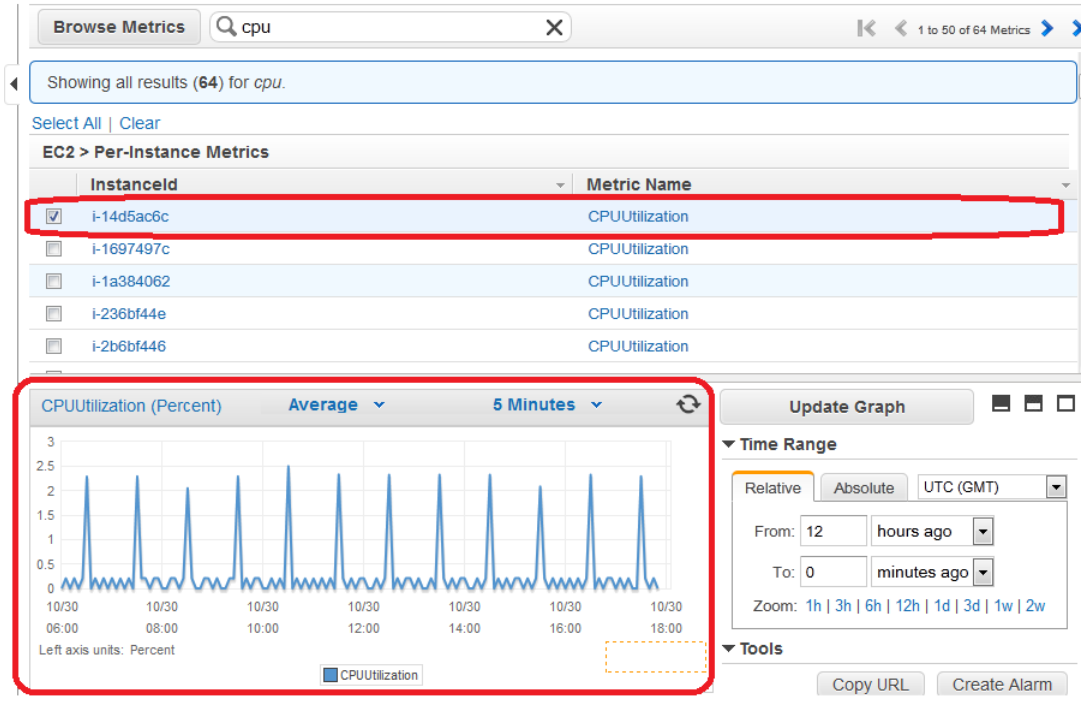
Graph a Metric

You can select a metric and create a graph of the data in CloudWatch. For example, you can select the CPUUtilization metric for an Amazon EC2 instance and display a graph of CPU usage over time for that instance.

To graph a metric

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see Regions and Endpoints in the *Amazon Web Services General Reference*.

3. In the navigation pane, click **Metrics**.
4. In the **CloudWatch Metrics by Category** pane, use the **Search Metrics** box and categories to find a metric by metric name, AWS resource, or other metadata.
5. Use the scroll bar and next and previous arrows above the metrics list to page through the full list of metrics
6. Select the metric to view, for example, CPUUtilization. A graph appears in the details pane.



7. To save this graph and access it later, in the details pane, under **Tools**, click **Copy URL**, and then in the **Copy Graph URL** dialog box, select the URL and paste it into your browser.

Graph a Metric Across Resources

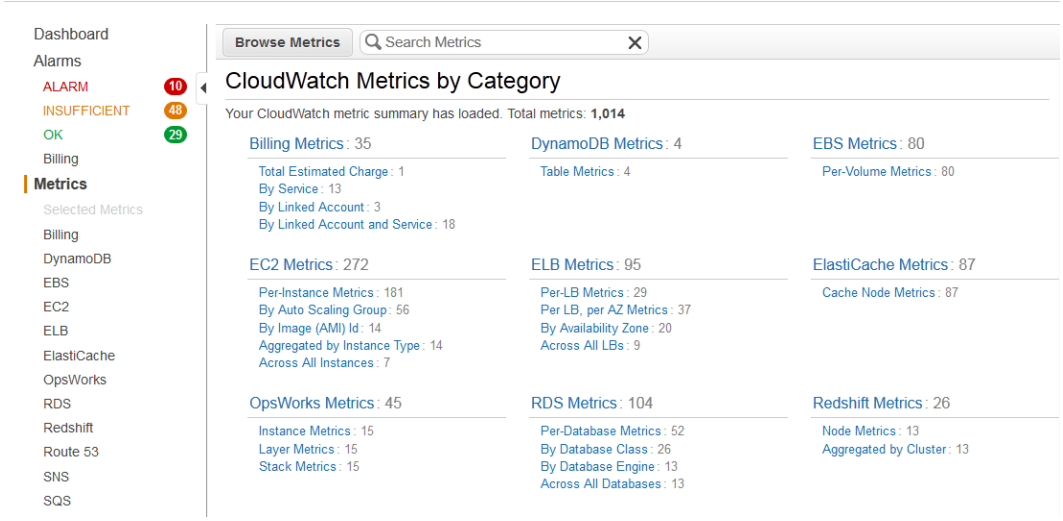
You can graph a metric across all resources to see everything on one graph. For example, you can graph the CPUUtilization metric for all Amazon EC2 instances on one graph.

To graph a metric across resources

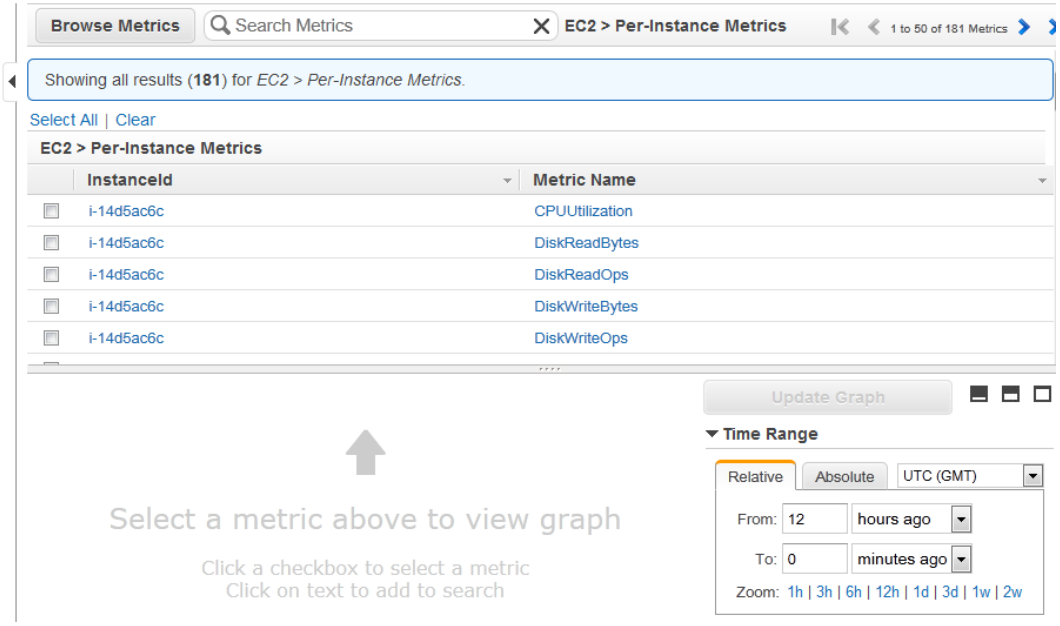
1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.

2. If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see Regions and Endpoints.

3. In the navigation pane, click **Metrics**.

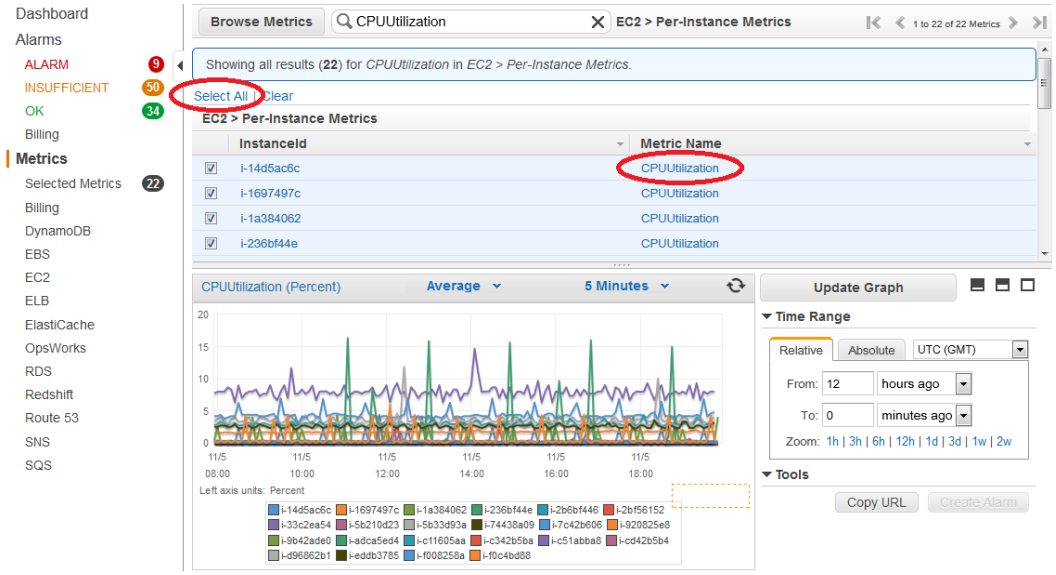


4. In the **CloudWatch Metrics by Category** pane, select a metric category. For example, under **EC2 Metrics**, select **Per-Instance Metrics**.



5. In the metric list, in the **Metric Name** column, click a metric. For example **CPUUtilization**.
6. At the top of the metric list, click **Select All**.

   The graph shows all data for all occurrences of the selected metric. In the example below, CPUUtilization for all Amazon EC2 instances is shown.



7. To save this graph and access it later, in the details pane, under **Tools**, click **Copy URL**, and then in the **Copy Graph URL** dialog box, select the URL and paste it into your browser.

Create a CloudWatch Alarm

You can create an Amazon CloudWatch alarm that monitors any one of your Amazon EC2 instance's CloudWatch metrics. CloudWatch will automatically send you a notification when the metric reaches a threshold you specify. You can create a CloudWatch alarm on the Amazon EC2 console of the AWS Management Console, or you can use the CloudWatch console and configure more advanced options.

Contents

- Send Email Based on CPU Usage Alarm (p. 361)
- Send Email Based on Load Balancer Alarm (p. 363)
- Send Email Based on Storage Throughput Alarm (p. 365)

Send Email Based on CPU Usage Alarm

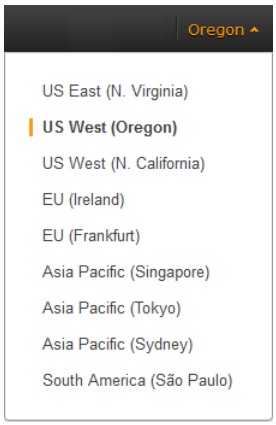
This scenario walks you through how to use the AWS Management Console or the command line interface to create an Amazon CloudWatch alarm that sends an Amazon Simple Notification Service email message when the alarm changes state from OK to ALARM.

In this scenario, you configure the alarm to change to the ALARM state when the average CPU use of an EC2 instance exceeds 70 percent for two consecutive five-minute periods.

AWS Management Console

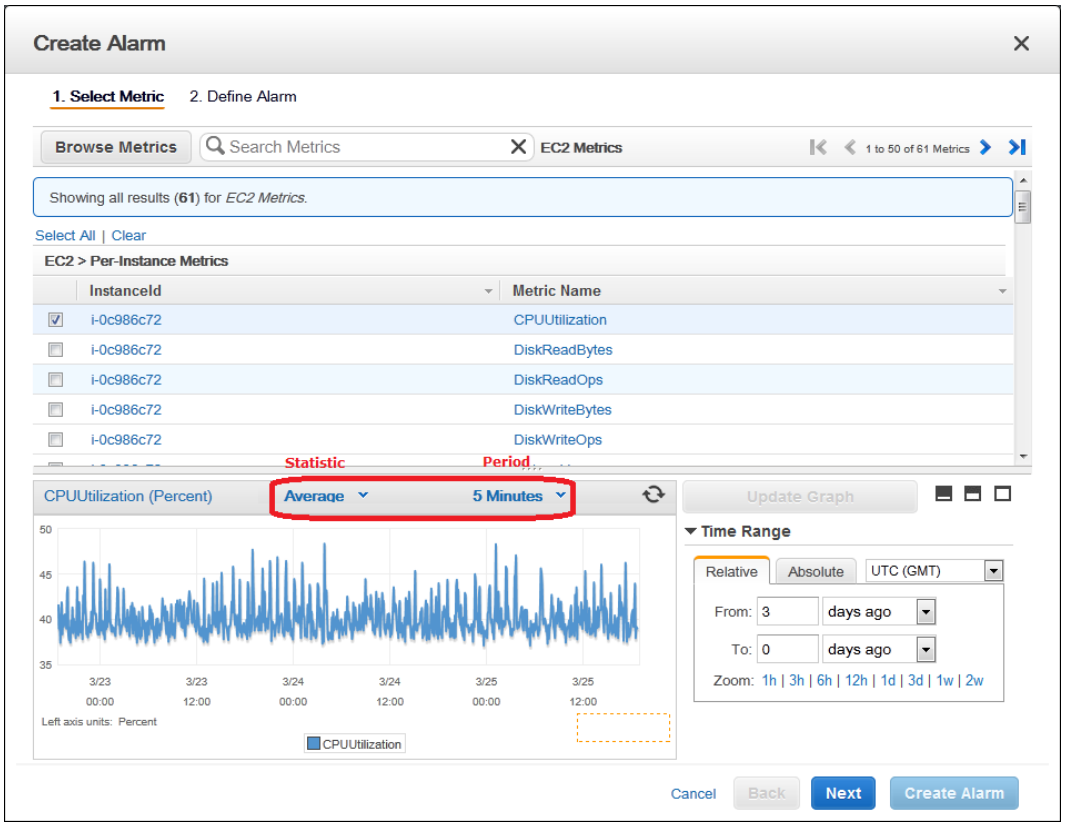
To create an alarm that sends email based on CPU usage

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see Regions and Endpoints.



3. In the navigation pane, click **Alarms**.
4. Click **Create Alarm**, and then in **CloudWatch Metrics by Category**, select a metric category, for example, **EC2 Metrics**.
5. In the list of metrics, select a row that contains **CPUUtilization** for a specific instance ID.

   A graph showing average `CPUUtilization` for a single instance appears in the lower pane.



6. Select **Average** from the **Statistic** drop-down list.
7. Select a period from the **Period** drop-down list, for example: **`5 minutes`**.
8. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: **`myHighCpuAlarm`**.
9. In the **Description** field, enter a description of the alarm, for example: **`CPU usage exceeds 70 percent.`**
10. In the **is** drop-down list, select **`>`**.
11. In the box next to the **is** drop-down list, enter **`70`** and in the **for** field, enter **`10`**.

    A graphical representation of the threshold is shown under **Alarm Preview**.
12. Under **Actions**, in the **Whenever this alarm** drop-down list, select **State is ALARM**.
13. In the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.
14. To create a new Amazon SNS topic, select **New list**.

    In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: **`myHighCpuAlarm`**, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the `ALARM` state.
15. Click **Create Alarm** to complete the alarm creation process.

Command Line Interface

To send an Amazon Simple Notification Service email message when CPU utilization exceeds 70 percent

1. Set up an Amazon Simple Notification Service topic or retrieve the Topic Resource Name of the topic you intend to use. For help on setting up an Amazon Simple Notification Service topic, see Set Up Amazon Simple Notification Service.
2. Create an alarm with the `put-metric-alarm` command. For more information about the `put-metric-alarm` command, see put-metric-alarm in the *AWS Command Line Interface Reference*. Use the values from the following example, but replace the values for `InstanceID` and `alarm-actions` with your own values.

```
C:\> aws cloudwatch
             put-metric-alarm --alarm-name cpu-mon --alarm-description
"Alarm when CPU exceeds 70%" --metric-name CPUUtilization --namespace AWS/EC2
 --statistic Average --period 300
             --threshold 70 --comparison-operator GreaterThanThreshold  -
-dimensions Name=InstanceId,Value=i-12345678 --evaluation-periods 2 --alarm-
actions arn:aws:sns:us-east-1:111122223333:MyTopic --unit Percent
```

   The AWS CLI returns to the command prompt if the command succeeds.
3. Test the alarm by forcing an alarm state change with the `set-alarm-state` command.

   a. Change the alarm state from `INSUFFICIENT_DATA` to `OK`:

   ```
   C:\> aws cloudwatch set-alarm-state  --alarm-name cpu-mon --state-reason
    "initializing" --state-value OK
   ```

   The AWS CLI returns to the command prompt if the command succeeds.

   b. Change the alarm state from `OK` to `ALARM`:

   ```
   C:\> aws cloudwatch set-alarm-state  --alarm-name cpu-mon --state-reason
    "initializing" --state-value ALARM
   ```

   The AWS CLI returns to the command prompt if the command succeeds.

   c. Check that an email has been received.

Send Email Based on Load Balancer Alarm

This scenario walks you through how to use the AWS Management Console or the command line interface to set up an Amazon Simple Notification Service notification and configure an alarm that monitors load balancer latency exceeding 100 ms.

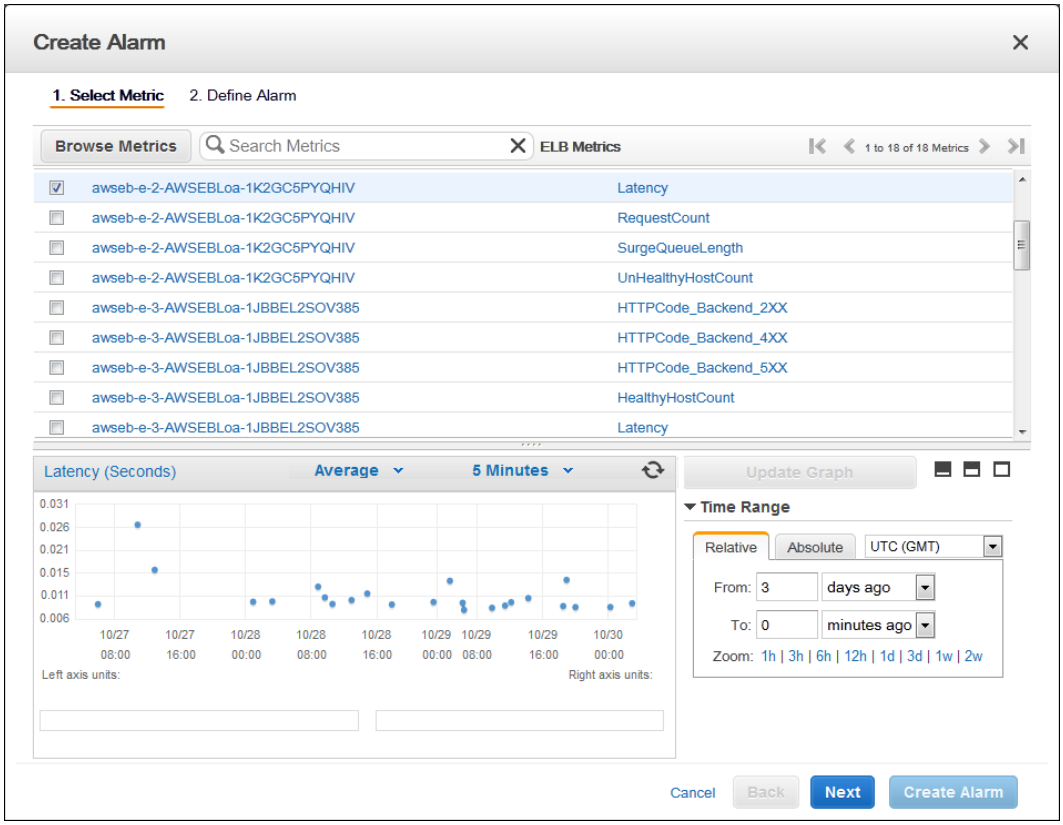
AWS Management Console

To create a load balancer alarm that sends email

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see Regions and Endpoints.

3. In the navigation pane, click **Alarms**.
4. Click **Create Alarm**, and then in the **CloudWatch Metrics by Category** pane, select a metric category, for example, **ELB Metrics**.
5. In the list of metrics, select a row that contains **Latency** for a specific load balancer.

   A graph showing average `Latency` for a single load balancer appears in the lower pane.



6. Select **Average** from the **Statistic** drop-down list.
7. Select **1 Minute** from the **Period** drop-down list.
8. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: **`myHighCpuAlarm`**.
9. In the **Description** field, enter a description of the alarm, for example: **`Alarm when Latency exceeds 100ms.`**

10. In the **is** drop-down list, select **>**.
11. In the box next to the **is** drop-down list, enter **0.1** and in the **for** field, enter **3**.

    A graphical representation of the threshold is shown under **Alarm Preview**.
12. Under **Actions**, in the **Whenever this alarm** drop-down list, select **State is ALARM**.
13. In the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.
14. To create a new Amazon SNS topic, select **New list**.

    In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: **myHighCpuAlarm**, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the ALARM state.
15. Click **Create Alarm** to complete the alarm creation process.

Command Line Interface

To send an Amazon Simple Notification Service email message when LoadBalancer Latency Exceeds 100 milliseconds

1. Create an Amazon Simple Notification Service topic. See instructions for creating an Amazon SNS topic in Set Up Amazon Simple Notification Service.
2. Use the `put-metric-alarm` command to create an alarm. For more information about the `put-metric-alarm` command, see put-metric-alarm in the *AWS Command Line Interface Reference*.

```
C:\> aws cloudwatch put-metric-alarm --alarm-name lb-mon --alarm-description
 "Alarm when Latency exceeds 100ms" --metric-name Latency --namespace AWS/ELB
 --statistic Average --period 60 --threshold 100 --comparison-operator
GreaterThanThreshold --dimensions Name=LoadBalancerName,Value=my-server --
evaluation-periods 3 --alarm-actions arn:aws:sns:us-east-1:1234567890:my-
topic --unit Milliseconds
```

   The AWS CLI returns to the command prompt if the command succeeds.
3. Test the alarm.

   - Force an alarm state change to ALARM:

```
C:\> aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason
"initializing" --state OK
C:\> aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason
"initializing" --state ALARM
```

     The AWS CLI returns to the command prompt if the command succeeds.
   - Check that an email has been received.

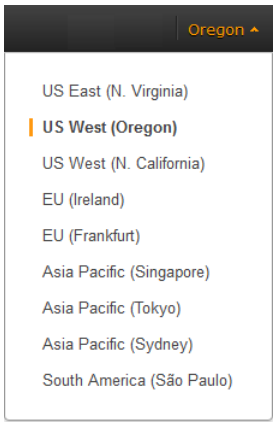
Send Email Based on Storage Throughput Alarm

This scenario walks you through how to use the AWS Management Console or the command line interface to set up an Amazon Simple Notification Service notification and to configure an alarm that sends email when EBS exceeds 100 MB throughput.

AWS Management Console

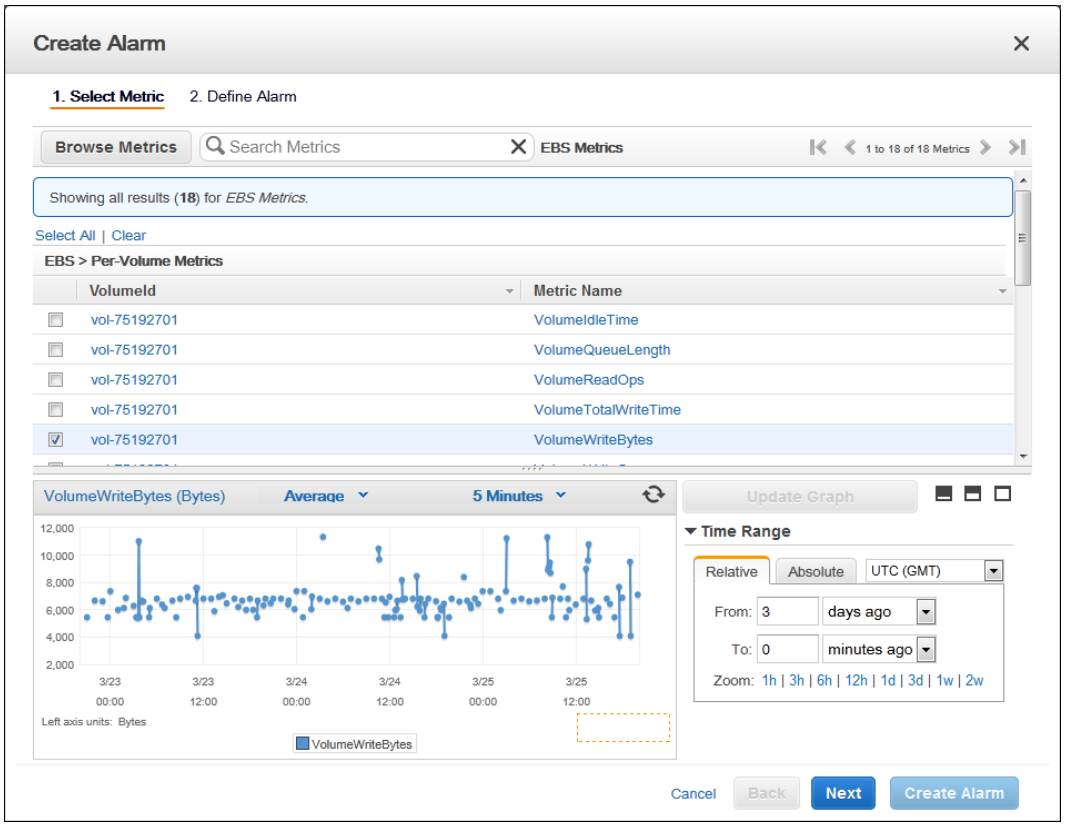
To create a storage throughput alarm that sends email

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see Regions and Endpoints.



3. In the navigation pane, click **Alarms**.
4. Click **Create Alarm**, and then in the **CloudWatch Metrics by Category** pane, select a metric category, for example, **EBS Metrics**.
5. In the list of metrics, select a row that contains **VolumeWriteBytes** for a specific VolumeId.

   A graph showing average `VolumeWriteBytes` for a single volume appears in the lower pane.



6. Select **Average** from the **Statistic** drop-down list.
7. Select **5 Minutes** from the **Period** drop-down list.
8. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: **`myHighWriteAlarm`**.
9. In the **Description** field, enter a description of the alarm, for example: **`VolumeWriteBytes exceeds 100,000 KiB/s.`**
10. In the **is** drop-down list, select **>**.
11. In the box next to the **is** drop-down list, enter **`100000`** and in the **for** field, enter **`15`**.

    A graphical representation of the threshold is shown under **Alarm Preview**.
12. Under **Actions**, in the **Whenever this alarm** drop-down list, select **State is ALARM**.
13. In the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.
14. To create a new Amazon SNS topic, select **New list**.

    In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: **`myHighCpuAlarm`**, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the `ALARM` state.
15. Click **Create Alarm** to complete the alarm creation process.

Command Line Interface

To send an Amazon Simple Notification Service email message when EBS exceeds 100 MB throughput

1. Create an Amazon Simple Notification Service topic. See instructions for creating an Amazon SNS topic in Set Up Amazon Simple Notification Service.
2. Use the `put-metric-alarm` command to create an alarm. For more information about the `put-metric-alarm` command, see put-metric-alarm in the *AWS Command Line Interface Reference*.

```
C:\> aws cloudwatch put-metric-alarm --alarm-name ebs-mon --alarm-description
 "Alarm when EBS volume exceeds 100MB throughput" --metric-name VolumeRead
Bytes --namespace AWS/EBS --statistic Average --period 300 --threshold
100000000 --comparison-operator GreaterThanThreshold --dimensions
Name=VolumeId,Value=my-volume-id --evaluation-periods 3 --alarm-actions
arn:aws:sns:us-east-1:1234567890:my-alarm-topic --insufficient-data-actions
 arn:aws:sns:us-east-1:1234567890:my-insufficient-data-topic
```

   The AWS CLI returns to the command prompt if the command succeeds.
3. Test the alarm.

   - Force an alarm state change to ALARM.

```
C:\> aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason
"initializing" --state-value OK
C:\> aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason
"initializing" --state-value ALARM
C:\> aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason
"initializing" --state-value INSUFFICIENT_DATA
```

   - Check that two emails have been received.

Create Alarms That Stop, Terminate, Reboot, or Recover an Instance

Using Amazon CloudWatch alarm actions, you can create alarms that automatically stop, terminate, reboot, or recover your Amazon Elastic Compute Cloud (Amazon EC2) instances. You can use the stop or terminate actions to help you save money when you no longer need an instance to be running. You can use the reboot and recover actions to automatically reboot those instances or recover them onto new hardware if a system impairment occurs.

Every alarm action you create uses alarm action ARNs. One set of ARNs is more secure because it requires you to have the EC2ActionsAccess IAM role in your account. This IAM role enables you to perform stop, terminate, or reboot actions--previously you could not execute an action if you were using an IAM role. Existing alarms that use the previous alarm action ARNs do not require this IAM role, however it is recommended that you change the ARN and add the role when you edit an existing alarm that uses these ARNs.

The EC2ActionsAccess IAM role enables AWS to perform alarm actions on your behalf. When you create an alarm action for the first time using the Amazon EC2 or Amazon CloudWatch consoles, AWS automatically creates this role for you. In addition, you must create the EC2ActionsAccess role using either console before it's available for use from the CLI.

There are a number of scenarios in which you might want to automatically stop or terminate your instance. For example, you might have instances dedicated to batch payroll processing jobs or scientific computing tasks that run for a period of time and then complete their work. Rather than letting those instances sit idle (and accrue charges), you can stop or terminate them which can help you to save money. The main difference between using the stop and the terminate alarm actions is that you can easily restart a stopped instance if you need to run it again later, and you can keep the same instance ID and root volume. However, you cannot restart a terminated instance. Instead, you must launch a new instance.

You can create an alarm that automatically recovers an Amazon EC2 instance when the instance becomes impaired due to an underlying hardware failure a problem that requires AWS involvement to repair. Examples of problems that cause system status checks to fail include:

- Loss of network connectivity
- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host

Important
The recover action is only supported on:

- C3, C4, M3, R3, and T2 instance types.
- Instances in the Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), EU (Ireland), EU (Frankfurt), South America (Sao Paulo), US East (N. Virginia), US West (N. California) and US West (Oregon) regions.
- Instances in a VPC.

  Note
  If your instance has a public IP address, it receives a new public IP address after recovery (if your subnet setting allows it). To retain the public IP address, use an Elastic IP address instead.
- Instances with shared tenancy (where the tenancy attribute of the instance is set to `default`).
- Instances that use Amazon EBS storage exclusively.

Currently, the recover action is not supported for EC2-Classic instances, dedicated tenancy instances, and instances that use any instance store volumes.

You can add the stop, terminate, reboot, or recover actions to any alarm that is set on an Amazon EC2 per-instance metric, including basic and detailed monitoring metrics provided by Amazon CloudWatch (in the AWS/EC2 namespace), as well as any custom metrics that include the “InstanceId=” dimension, as long as the InstanceId value refers to a valid running Amazon EC2 instance.

Contents

- Adding Stop Actions to Amazon CloudWatch Alarms (p. 369)
- Adding Terminate Actions to Amazon CloudWatch Alarms (p. 372)
- Adding Reboot Actions to Amazon CloudWatch Alarms (p. 375)
- Adding Recover Actions to Amazon CloudWatch Alarms (p. 377)
- Using the Amazon CloudWatch Console to View the History of Triggered Alarms and Actions (p. 381)
- Using the CLI or the API to Create an Alarm to Stop, Terminate, Reboot, or Recover an Instance (p. 382)
- Amazon CloudWatch Alarm Action Scenarios (p. 387)

Adding Stop Actions to Amazon CloudWatch Alarms

You can configure the stop alarm action using the Amazon EC2 console, the Amazon CloudWatch console, the Amazon CloudWatch command line interface (CLI), the CloudWatch API, or the AWS SDKs. For information about using the Amazon CloudWatch API with the AWS SDKs, see Sample Code & Libraries.

Using the Amazon EC2 Console to Create an Alarm to Stop an Instance

You can create an alarm that stops an Amazon EC2 instance when a certain threshold has been met. For example, you may run development or test instances and occasionally forget to shut them off. You can create an alarm that is triggered when the average CPU utilization percentage has been lower than 10 percent for 24 hours, signaling that it is idle and no longer in use. You can adjust the threshold, duration, and period to suit your needs, plus you can add an Amazon Simple Notification Service (Amazon SNS) notification, so that you will receive an email when the alarm is triggered.

Amazon EC2 instances that use an Amazon Elastic Block Store volume as the root device can be stopped or terminated, whereas instances that use the instance store as the root device can only be terminated.

Note

If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions:

- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` for all alarms on Amazon EC2 instance status metrics.
- `ec2:StopInstances` for alarms with stop actions.
- `ec2:TerminateInstances` for alarms with terminate actions.
- `ec2:DescribeInstanceRecoveryAttribute`, and `ec2:RecoverInstances` for alarms with recover actions.

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see Permissions and Policies in the *IAM User Guide*.

If you want to use an IAM role to stop or terminate an instance using an alarm action, you can only use the EC2ActionsAccess role. Other IAM roles are not supported. If you are using another

IAM role, you cannot stop or terminate the instance. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.
If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot recover an Amazon EC2 instance using alarm actions.

To create an alarm to stop an idle instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see Regions and Endpoints.
3. In the navigation pane, under **INSTANCES**, click **Instances**.
4. In the contents pane, right-click an instance, select **CloudWatch Monitoring**, and then click **Add/Edit Alarms**.

   Or, you can also select the instance, and then in the lower pane on the **Monitoring** tab, click **Create Alarm**.
5. In the **Alarm Details for** dialog box, click **Create Alarm**.
6. If you want to receive an email when the alarm is triggered, in the **Create Alarm for** dialog box, in the **Send a notification to** box, select an existing Amazon SNS topic, or click **Create Topic** to create a new one.

   If you create a new topic, in the **Send a notification to** box type a name for the topic, and then in the **With these recipients** box, type the email addresses of the recipients (separated by commas). Later, after you create the alarm, you will receive a subscription confirmation email that you must accept before you will get email for this topic.
7. Select the **Take the action** check box, and then choose the **Stop this instance** radio button.
8. If prompted, select the **Create IAM role: EC2ActionsAccess** check box to automatically create an IAM role so that AWS can automatically stop the instance on your behalf when the alarm is triggered.
9. In the **Whenever** boxes, choose the statistic you want to use and then select the metric. In this example, choose **Average** and **CPU Utilization**.
10. In the **Is** boxes, define the metric threshold. In this example, enter **10** percent.
11. In the **For at least** box, choose the sampling period for the alarm. In this example, enter **24** consecutive periods of one hour.
12. To change the name of the alarm, in the **Name this alarm** box, type a new name.

    If you don't type a name for the alarm, Amazon CloudWatch will automatically create one for you.

    Note
    You can adjust the alarm configuration based on your own requirements before creating the alarm, or you can edit them later. This includes the metric, threshold, duration, action, and notification settings. However, after you create an alarm, you cannot edit its name later.
13. Click **Create Alarm**.

Using the Amazon CloudWatch Console to Create an Alarm that Stops an Instance

You can create an alarm that stops an Amazon EC2 instance when a certain threshold has been met. For example, you may run development or test instances and occasionally forget to shut them off. You can create an alarm that is triggered when the average CPU utilization percentage has been lower than 10 percent for 24 hours, signaling that it is idle and no longer in use. You can adjust the threshold, duration, and period to suit your needs, plus you can add an Amazon Simple Notification Service (Amazon SNS) notification, so that you will receive an email when the alarm is triggered.

Amazon CloudWatch alarm actions can stop an EBS-backed Amazon EC2 instances but they cannot stop instance store-backed Amazon EC2 instances. However, Amazon CloudWatch alarm actions can terminate either type of Amazon EC2 instance.

Note

If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions:

- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` for all alarms on Amazon EC2 instance status metrics.
- `ec2:StopInstances` for alarms with stop actions.
- `ec2:TerminateInstances` for alarms with terminate actions.
- `ec2:DescribeInstanceRecoveryAttribute`, and `ec2:RecoverInstances` for alarms with recover actions.

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see Permissions and Policies in the *IAM User Guide*.
If you want to use an IAM role to stop or terminate an instance using an alarm action, you can only use the EC2ActionsAccess role. Other IAM roles are not supported. If you are using another IAM role, you cannot stop or terminate the instance. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.
If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot recover an Amazon EC2 instance using alarm actions.

To create an alarm to stop an idle instance

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see Regions and Endpoints.
3. In the navigation pane, click **Alarms**.
4. Click **Create Alarm**, and then in **CloudWatch Metrics by Category**, under **EC2 Metrics**, select **Per-Instance Metrics**.
5. In the list of metrics, select the instance and metric you want to create an alarm for. You can also type an instance ID in the search box to go the instance that you want.
6. Select **Average** from the **Statistic** drop-down list.
7. Select a period from the **Period** drop-down list, for example: **`1 Day`**.
8. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: **`Stop EC2 instance`**.
9. In the **Description** field, enter a description of the alarm, for example: **`Stop EC2 instance when CPU is idle for too long.`**
10. In the **is** drop-down list, select **<**.
11. In the box next to the **is** drop-down list, enter **`10`** and in the **for** field, enter **`1440`**.

    A graphical representation of the threshold is shown under **Alarm Preview**.
12. Under **Actions**, click **EC2 Action**.
13. In the **Whenever this alarm** drop-down list, select **State is ALARM**.
14. In the **Take this action** drop-down list, select **Stop this instance**.
15. If prompted, select the **Create IAM role: EC2ActionsAccess** check box to automatically create an IAM role so that AWS can automatically stop the instance on your behalf when the alarm is triggered.

16. Click **Notification**, and then in the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.
17. To create a new Amazon SNS topic, select **New list**.

    In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: **`Stop_EC2_Instance`**, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the `ALARM` state.

    Important
    If you are creating a new topic or adding email addresses to an existing topic, each email address that you add will be sent a topic subscription confirmation email. You must confirm the subscription by clicking the included link before notifications will be sent to a new email address.
18. In the navigation pane, click **Create Alarm** to complete the alarm creation process.

Adding Terminate Actions to Amazon CloudWatch Alarms

You can configure the terminate alarm action using the Amazon EC2 console, the Amazon CloudWatch console, the Amazon CloudWatch command line interface (CLI), the CloudWatch API, or the AWS SDKs. For information about using the Amazon CloudWatch API with the AWS SDKs, see Sample Code & Libraries.

Using the Amazon EC2 Console to Create an Alarm that Terminates an Instance

You can create an alarm that terminates an EC2 instance automatically when a certain threshold has been met (as long as termination protection is not enabled for the instance). For example, you might want to terminate an instance when it has completed its work, and you don't need the instance again. If you might want to use the instance later, you should stop the instance instead of terminating it. For information on enabling and disabling termination protection for an instance, see Enabling Termination Protection for an Instance in the *Amazon EC2 User Guide for Linux Instances*.

Note
If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions:

- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` for all alarms on Amazon EC2 instance status metrics.
- `ec2:StopInstances` for alarms with stop actions.
- `ec2:TerminateInstances` for alarms with terminate actions.
- `ec2:DescribeInstanceRecoveryAttribute`, and `ec2:RecoverInstances` for alarms with recover actions.

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see Permissions and Policies in the *IAM User Guide*.
If you want to use an IAM role to stop or terminate an instance using an alarm action, you can only use the EC2ActionsAccess role. Other IAM roles are not supported. If you are using another IAM role, you cannot stop or terminate the instance. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.
If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot recover an Amazon EC2 instance using alarm actions.

To create an alarm to terminate an idle instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see Regions and Endpoints.
3. In the navigation pane, under **INSTANCES**, click **Instances**.
4. In the upper pane, right-click an instance, select **CloudWatch Monitoring**, and then click **Add/Edit Alarms**.

   Or, select the instance and then in the lower pane, on the **Monitoring** tab, click **Create Alarm**.
5. In the **Alarm Details for** dialog box, click **Create Alarm**.
6. If you want to receive an email when the alarm is triggered, in the **Create Alarm for** dialog box, in the **Send a notification to** box, select an existing Amazon SNS topic, or click **Create Topic** to create a new one.

   If you create a new topic, in the **Send a notification to** box type a name for the topic, and then in the **With these recipients** box, type the email addresses of the recipients (separated by commas). Later, after you create the alarm, you will receive a subscription confirmation email that you must accept before you will get email for this topic.
7. Select the **Take the action** check box, and then choose the **Terminate this instance** radio button.
8. If prompted, select the **Create IAM role: EC2ActionsAccess** check box to automatically create an IAM role so that AWS can automatically stop the instance on your behalf when the alarm is triggered.
9. In the **Whenever** boxes, choose the statistic you want to use and then select the metric. In this example, choose **Average** and **CPU Utilization**.
10. In the **Is** boxes, define the metric threshold. In this example, enter **10** percent.
11. In the **For at least** box, choose the sampling period for the alarm. In this example, enter **24** consecutive periods of one hour.
12. To change the name of the alarm, in the **Name this alarm** box, type a new name.

    If you don't type a name for the alarm, Amazon CloudWatch will automatically create one for you.

    Note
    You can adjust the alarm configuration based on your own requirements before creating the alarm, or you can edit them later. This includes the metric, threshold, duration, action, and notification settings. However, after you create an alarm, you cannot edit its name later.
13. Click **Create Alarm**.

Using the Amazon CloudWatch Console to Create an Alarm to Terminate an Idle Instance

You can create an alarm that terminates an Amazon EC2 instance automatically when a certain threshold has been met, as long as termination protection is disabled on the instance. For example, you might want to terminate an instance when it has completed its work, and you don't need the instance again. If you might want to use the instance later, you should stop the instance instead of terminating it. For information on disabling termination protection on an instance, see Enabling Termination Protection for an Instance in the *Amazon EC2 User Guide for Linux Instances*.

Note
If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions:

- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` for all alarms on Amazon EC2 instance status metrics.
- `ec2:StopInstances` for alarms with stop actions.

- `ec2:TerminateInstances` for alarms with terminate actions.
- `ec2:DescribeInstanceRecoveryAttribute`, and `ec2:RecoverInstances` for alarms with recover actions.

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see Permissions and Policies in the *IAM User Guide*.
If you want to use an IAM role to stop or terminate an instance using an alarm action, you can only use the EC2ActionsAccess role. Other IAM roles are not supported. If you are using another IAM role, you cannot stop or terminate the instance. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.
If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot recover an Amazon EC2 instance using alarm actions.

To create an alarm to terminate an idle instance

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see Regions and Endpoints.
3. In the navigation pane, click **Alarms**.
4. Click **Create Alarm**, and then in **CloudWatch Metrics by Category**, under **EC2 Metrics**, select **Per-Instance Metrics**.
5. In the list of metrics, select the instance and metric you want to create an alarm for. You can also type an instance ID in the search box to go the instance that you want.
6. Select **Average** from the **Statistic** drop-down list.
7. Select a period from the **Period** drop-down list, for example: **`1 Day`**.
8. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: **`Terminate EC2 instance`**.
9. In the **Description** field, enter a description of the alarm, for example: **`Terminate EC2 instance when CPU is idle for too long.`**
10. In the **is** drop-down list, select **<**.
11. In the box next to the **is** drop-down list, enter **`10`** and in the **for** field, enter **`1440`**.

    A graphical representation of the threshold is shown under **Alarm Preview**.
12. Under **Actions**, click EC2 Action.
13. In the **Whenever this alarm** drop-down list, select **State is ALARM**.
14. In the **Take this action** drop-down list, select **Terminate this instance**.
15. If prompted, select the **Create IAM role: EC2ActionsAccess** check box to automatically create an IAM role so that AWS can automatically stop the instance on your behalf when the alarm is triggered.
16. Click **Notification**, and then in the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.
17. To create a new Amazon SNS topic, select **New list**.

    In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: **`Terminate_EC2_Instance`**, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the `ALARM` state.

    Important
    If you are creating a new topic or adding email addresses to an existing topic, each email address that you add will be sent a topic subscription confirmation email. You must confirm the subscription by clicking the included link before notifications will be sent to a new email address.

18. In the navigation pane, click **Create Alarm** to complete the alarm creation process.

Adding Reboot Actions to Amazon CloudWatch Alarms

You can configure the reboot alarm action using the Amazon EC2 console, the Amazon CloudWatch console, the Amazon CloudWatch command line interface (CLI), the CloudWatch API, or the AWS SDKs. For information about using the Amazon CloudWatch API with the AWS SDKs, see Sample Code & Libraries.

Using the Amazon EC2 Console to Create an Alarm to Reboot an Instance

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically reboots the instance. The reboot alarm action is recommended for Instance Health Check failures (as opposed to the recover alarm action, which is suited for System Health Check failures). An instance reboot is equivalent to an operating system reboot. In most cases, it takes only a few minutes to reboot your instance. When you reboot an instance, it remains on the same physical host, so your instance keeps its public DNS name, private IP address, and any data on its instance store volumes.

Rebooting an instance doesn't start a new instance billing hour, unlike stopping and restarting your instance. For more information about rebooting an instance, see Reboot Your Instance in the *Amazon EC2 User Guide for Linux Instances*.

Note
If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions:

- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` for all alarms on Amazon EC2 instance status metrics.
- `ec2:StopInstances` for alarms with stop actions.
- `ec2:TerminateInstances` for alarms with terminate actions.
- `ec2:DescribeInstanceRecoveryAttribute`, and `ec2:RecoverInstances` for alarms with recover actions.

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see Permissions and Policies in the *IAM User Guide*.
If you want to use an IAM role to stop or terminate an instance using an alarm action, you can only use the EC2ActionsAccess role. Other IAM roles are not supported. If you are using another IAM role, you cannot stop or terminate the instance. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.
If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot recover an Amazon EC2 instance using alarm actions.

To create an alarm to reboot an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see Regions and Endpoints.
3. In the navigation pane, under **INSTANCES**, click **Instances**.
4. In the upper pane, right-click an instance, select **CloudWatch Monitoring**, and then click **Add/Edit Alarms**.

   Or, select the instance and then in the lower pane, on the **Monitoring** tab, click **Create Alarm**.

5. In the **Alarm Details for** dialog box, click **Create Alarm**.
6. If you want to receive an email when the alarm is triggered, in the **Create Alarm for** dialog box, in the **Send a notification to** box, select an existing Amazon SNS topic, or click **Create Topic** to create a new one.

   If you create a new topic, in the **Send a notification to** box type a name for the topic, and then in the **With these recipients** box, type the email addresses of the recipients (separated by commas). Later, after you create the alarm, you will receive a subscription confirmation email that you must accept before you will get email for this topic.
7. Select the **Take the action** check box, and then choose the **Reboot this instance** radio button.
8. If prompted, select the **Create IAM role: EC2ActionsAccess** check box to automatically create an IAM role so that AWS can automatically stop the instance on your behalf when the alarm is triggered.
9. In the **Whenever** box, choose Status Check Failed (Instance).
10. In the **For at least** field, enter **2**.
11. In the **consecutive period(s) of** box, select **1 minute**.
12. To change the name of the alarm, in the **Name of alarm** box, type a new name.

    If you don't type a name for the alarm, Amazon CloudWatch will automatically create one for you.
13. Click **Create Alarm**.

Using the Amazon CloudWatch Console to Create an Alarm to Reboot an Instance

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically reboots the instance. The reboot alarm action is recommended for Instance Health Check failures (as opposed to the recover alarm action, which is suited for System Health Check failures). An instance reboot is equivalent to an operating system reboot. In most cases, it takes only a few minutes to reboot your instance. When you reboot an instance, it remains on the same physical host, so your instance keeps its public DNS name, private IP address, and any data on its instance store volumes.

Rebooting an instance doesn't start a new instance billing hour, unlike stopping and restarting your instance. For more information about rebooting an instance, see Reboot Your Instance in the *Amazon EC2 User Guide for Linux Instances*.

Note

If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions:

- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` for all alarms on Amazon EC2 instance status metrics.
- `ec2:StopInstances` for alarms with stop actions.
- `ec2:TerminateInstances` for alarms with terminate actions.
- `ec2:DescribeInstanceRecoveryAttribute`, and `ec2:RecoverInstances` for alarms with recover actions.

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see Permissions and Policies in the *IAM User Guide*.
If you want to use an IAM role to stop or terminate an instance using an alarm action, you can only use the EC2ActionsAccess role. Other IAM roles are not supported. If you are using another IAM role, you cannot stop or terminate the instance. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.

If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot recover an Amazon EC2 instance using alarm actions.

To create an alarm to reboot an instance

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see Regions and Endpoints.
3. In the navigation pane, click **Alarms**.
4. Click **Create Alarm**, and then in **CloudWatch Metrics by Category**, under **EC2 Metrics**, select **Per-Instance Metrics**.
5. In the list of metrics, select the instance and `StatusCheckFailed_Instance` metric you want to create an alarm for. You can also type an instance ID in the search box to go the instance that you want.
6. Select **Minimum** from the **Statistic** drop-down list.

   Note
   This is the only statistic that is currently supported.
7. Select a period from the **Period** drop-down list, for example: **`1 Minute`**.
8. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: **`Reboot EC2 instance`**.
9. In the **Description** field, enter a description of the alarm, for example: **`Reboot EC2 instance when health checks fail.`**
10. In the **is** drop-down list, select **>**.
11. In the box next to the **is** drop-down list, enter **`0`** and in the **for** field, enter **`2`**.

    A graphical representation of the threshold is shown under **Alarm Preview**.
12. Under **Actions**, click EC2 Action.
13. In the **Whenever this alarm** drop-down list, select **State is ALARM**.
14. In the **Take this action** drop-down list, select **Reboot this instance**.
15. Click **Notification**, and then in the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.
16. To create a new Amazon SNS topic, select **New list**.

    In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: **`Reboot_EC2_Instance`**, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the `ALARM` state.

    Important
    If you are creating a new topic or adding email addresses to an existing topic, each email address that you add will be sent a topic subscription confirmation email. You must confirm the subscription by clicking the included link before notifications will be sent to a new email address.
17. In the navigation pane, click **Create Alarm** to complete the alarm creation process.

Adding Recover Actions to Amazon CloudWatch Alarms

You can configure the recover alarm action using the Amazon EC2 console, the Amazon CloudWatch console, the Amazon CloudWatch command line interface (CLI), the CloudWatch API, or the AWS SDKs. For information about using the Amazon CloudWatch API with the AWS SDKs, see Sample Code & Libraries.

Using the Amazon EC2 Console to Create an Alarm to Recover an Instance

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair. A recovered instance is identical to the original instance, including the instance ID, private IP addresses, Elastic IP addresses, and all instance metadata.

When the `StatusCheckFailed_System` alarm is triggered, and the recover action is initiated, you will be notified by the Amazon SNS topic that you selected when you created the alarm and associated the recover action. During instance recovery, the instance is migrated during an instance reboot, and any data that is in-memory is lost. When the process is complete, you'll receive an email notification that includes the status of the recovery attempt and any further instructions. You will notice an instance reboot on the recovered instance.

Examples of problems that cause system status checks to fail include:

- Loss of network connectivity
- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host

Important
The recover action is only supported on:

- C3, C4, M3, R3, and T2 instance types.
- Instances in the Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), EU (Ireland), EU (Frankfurt), South America (Sao Paulo), US East (N. Virginia), US West (N. California) and US West (Oregon) regions.
- Instances in a VPC. Dedicated instances are not supported.

  Note
  If your instance has a public IP address, it receives a new public IP address after recovery. To retain the public IP address, use an Elastic IP address instead.
- Instances that use EBS-backed storage. Instance storage is not supported. Automatic recovery of the instance will fail if any instance storage is attached.

Note
If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions:

- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` for all alarms on Amazon EC2 instance status metrics.
- `ec2:StopInstances` for alarms with stop actions.
- `ec2:TerminateInstances` for alarms with terminate actions.
- `ec2:DescribeInstanceRecoveryAttribute`, and `ec2:RecoverInstances` for alarms with recover actions.

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see Permissions and Policies in the *IAM User Guide*.
If you want to use an IAM role to stop or terminate an instance using an alarm action, you can only use the EC2ActionsAccess role. Other IAM roles are not supported. If you are using another

IAM role, you cannot stop or terminate the instance. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.
If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot recover an Amazon EC2 instance using alarm actions.

To create an alarm to recover an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see Regions and Endpoints.
3. In the navigation pane, under **INSTANCES**, click **Instances**.
4. In the upper pane, right-click an instance, select **CloudWatch Monitoring**, and then click **Add/Edit Alarms**.

   Or, select the instance and then in the lower pane, on the **Monitoring** tab, click **Create Alarm**.
5. In the **Alarm Details for** dialog box, click **Create Alarm**.
6. If you want to receive an email when the alarm is triggered, in the **Create Alarm for** dialog box, in the **Send a notification to** box, select an existing Amazon SNS topic, or click **Create Topic** to create a new one.

   If you create a new topic, in the **Send a notification to** box type a name for the topic, and then in the **With these recipients** box, type the email addresses of the recipients (separated by commas). Later, after you create the alarm, you will receive a subscription confirmation email that you must accept before you will get email for this topic.
7. Select the **Take the action** check box, and then choose the **Recover this instance** radio button.
8. If prompted, select the **Create IAM role: EC2ActionsAccess** check box to automatically create an IAM role so that AWS can automatically stop the instance on your behalf when the alarm is triggered.
9. In the **Whenever** box, choose Status Check Failed (System).
10. In the **For at least** field, enter **2**.
11. In the **consecutive period(s) of** box, select **1 minute**.
12. To change the name of the alarm, in the **Name of alarm** box, type a new name.

    If you don't type a name for the alarm, Amazon CloudWatch will automatically create one for you.
13. Click **Create Alarm**.

Using the Amazon CloudWatch Console to Create an Alarm to Recover an Instance

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair. A recovered instance is identical to the original instance, including the instance ID, private IP addresses, Elastic IP addresses, and all instance metadata.

When the `StatusCheckFailed_System` alarm is triggered, and the recover action is initiated, you will be notified by the Amazon SNS topic that you selected when you created the alarm and associated the recover action. During instance recovery, the instance is migrated during an instance reboot, and any data that is in-memory is lost. When the process is complete, you'll receive an email notification that includes the status of the recovery attempt and any further instructions. You will notice an instance reboot on the recovered instance.

Examples of problems that cause system status checks to fail include:

- Loss of network connectivity
- Loss of system power

- Software issues on the physical host
- Hardware issues on the physical host

Important
The recover action is only supported on:

- C3, C4, M3, R3, and T2 instance types.
- Instances in the Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), EU (Ireland), EU (Frankfurt), South America (Sao Paulo), US East (N. Virginia), US West (N. California) and US West (Oregon) regions.
- Instances in a VPC. Dedicated instances are not supported.

  Note
  If your instance has a public IP address, it receives a new public IP address after recovery. To retain the public IP address, use an Elastic IP address instead.
- Instances that use EBS-backed storage. Instance storage is not supported. Automatic recovery of the instance will fail if any instance storage is attached.

Note
If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions:

- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` for all alarms on Amazon EC2 instance status metrics.
- `ec2:StopInstances` for alarms with stop actions.
- `ec2:TerminateInstances` for alarms with terminate actions.
- `ec2:DescribeInstanceRecoveryAttribute`, and `ec2:RecoverInstances` for alarms with recover actions.

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see Permissions and Policies in the *IAM User Guide*.
If you want to use an IAM role to stop or terminate an instance using an alarm action, you can only use the EC2ActionsAccess role. Other IAM roles are not supported. If you are using another IAM role, you cannot stop or terminate the instance. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.
If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot recover an Amazon EC2 instance using alarm actions.

To create an alarm to recover an instance

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see Regions and Endpoints.
3. In the navigation pane, click **Alarms**.
4. Click **Create Alarm**, and then in **CloudWatch Metrics by Category**, under **EC2 Metrics**, select **Per-Instance Metrics**.
5. In the list of metrics, select the instance and `StatusCheckFailed_System` metric you want to create an alarm for. You can also type an instance ID in the search box to go the instance that you want.
6. Select **Minimum** from the **Statistic** drop-down list.

Note
This is the only statistic that is currently supported.

7. Select a period from the **Period** drop-down list, for example: **`1 Minute`**.
8. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: **`Recover EC2 instance`**.
9. In the **Description** field, enter a description of the alarm, for example: **`Recover EC2 instance when health checks fail.`**
10. In the **is** drop-down list, select **`>`**.
11. In the box next to the **is** drop-down list, enter **`0`** and in the **for** field, enter **`2`**.

    A graphical representation of the threshold is shown under **Alarm Preview**.
12. Under **Actions**, click EC2 Action.
13. In the **Whenever this alarm** drop-down list, select **State is ALARM**.
14. In the **Take this action** drop-down list, select **Recover this instance**.
15. Click **Notification**, and then in the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.
16. To create a new Amazon SNS topic, select **New list**.

    In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: **`Recover_EC2_Instance`**, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the `ALARM` state.

    Important
    If you are creating a new topic or adding email addresses to an existing topic, each email address that you add will be sent a topic subscription confirmation email. You must confirm the subscription by clicking the included link before notifications will be sent to a new email address.
17. In the navigation pane, click **Create Alarm** to complete the alarm creation process.

Using the Amazon CloudWatch Console to View the History of Triggered Alarms and Actions

You can view alarm and action history in the Amazon CloudWatch console. Amazon CloudWatch keeps the last two weeks' worth of alarm and action history.

To view the history of triggered alarms and actions

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see Regions and Endpoints.
3. In the navigation pane, click **Alarms**.
4. In the upper pane, select the alarm with the history that you want to view.
5. In the lower pane, the **Details** tab shows the most recent state transition along with the time and metric values.
6. Click the **History** tab to view the most recent history entries.

Using the CLI or the API to Create an Alarm to Stop, Terminate, Reboot, or Recover an Instance

If you are using either the AWS CLI or the Amazon CloudWatch API, or if you are using the AWS SDKs with the API, you can create a CloudWatch alarm using an Amazon EC2 per-instance metric, and then add an action using the action's dedicated Amazon Resource Name (ARN). You can add the action to any alarm state, and you can specify the region for each action. The region must match the region to which you send the put-metric-alarm request.

Action	ARN (with region)	ARN (for use with IAM role)
Stop	arn:aws:automate:us-east-1:ec2:stop	arn:aws:swf:us-east-1:{*customer-account*}:action/actions/AWS_EC2.InstanceId.Stop/1.0 **Note** You must create at least one stop alarm using the Amazon EC2 or CloudWatch console to create the **EC2ActionsAccess** IAM role. After this IAM role is created, you can create stop alarms using the CLI.
Terminate	arn:aws:automate:us-east-1:ec2:terminate	arn:aws:swf:us-east-1:{*customer-account*}:action/actions/AWS_EC2.InstanceId.Terminate/1.0 **Note** You must create at least one terminate alarm using the Amazon EC2 or CloudWatch console to create the **EC2ActionsAccess** IAM role. After this IAM role is created, you can create terminate alarms using the CLI.
Reboot	n/a	arn:aws:swf:us-east-1:{*customer-account*}:action/actions/AWS_EC2.InstanceId.Reboot/1.0 **Note** You must create at least one reboot alarm using the Amazon EC2 or CloudWatch console to create the **EC2ActionsAccess** IAM role. After this IAM role is created, you can create reboot alarms using the CLI.

Action	ARN (with region)	ARN (for use with IAM role)
Recover	arn:aws:automate:us-east-1:ec2:recover	n/a

For information about using the Amazon CloudWatch API with the AWS SDKs, see Sample Code & Libraries.

Note
If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions:

- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` for all alarms on Amazon EC2 instance status metrics.
- `ec2:StopInstances` for alarms with stop actions.
- `ec2:TerminateInstances` for alarms with terminate actions.
- `ec2:DescribeInstanceRecoveryAttribute`, and `ec2:RecoverInstances` for alarms with recover actions.

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see Permissions and Policies in the *IAM User Guide*.
If you want to use an IAM role to stop or terminate an instance using an alarm action, you can only use the EC2ActionsAccess role. Other IAM roles are not supported. If you are using another IAM role, you cannot stop or terminate the instance. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.
If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot recover an Amazon EC2 instance using alarm actions.

To create an alarm to stop an instance using the CLI

You can use the `arn:aws:automate:us-east-1:ec2:stop` ARN to stop an Amazon EC2 instance. The following example shows how to stop an instance if the average CPU utilization is less than 10 percent over a 24 hour period.

- At a command prompt, type:

```
% aws cloudwatch put-metric-alarm --alarm-name my-Alarm --alarm-description
 "Stop the instance when it is idle for a day" --namespace "AWS/EC2" --di
mensions Name=InstanceId,Value="i-abc123" --statistic Average --metric-name
 CPUUtilization --comparison-operator LessThanThreshold --threshold 10 --
period 86400 --evaluation-periods 4 --alarm-actions arn:aws:automate:us-
east-1:ec2:stop
```

To create an alarm to terminate an instance using the CLI

- At a command prompt, type:

```
% aws cloudwatch put-metric-alarm --alarm-name my-Alarm --alarm-description
 "Terminate the instance when it is idle for a day" --namespace "AWS/EC2"
--dimensions Name=InstanceId,Value="i-abc123" --statistic Average  --metric-
```

```
name CPUUtilization --comparison-operator LessThanThreshold --threshold 1
--period 86400 --evaluation-periods 4 -- alarm-actions arn:aws:automate:us-
east-1:ec2:terminate
```

To create an alarm to reboot an instance using the CLI

- At a command prompt, type:

```
% aws cloudwatch put-metric-alarm --alarm-name my-Alarm --alarm-description
 "Reboot the instance" --namespace "AWS/EC2" --dimensions Name=In
stanceId,Value="i-abc123" --statistic Minimum  --metric-name StatusCheck
Failed_Instance --comparison-operator GreaterThanThreshold --threshold 0 -
-period 60 --evaluation-periods 2 --alarm-actions arn:aws:swf:us-east-
1:{customer-account}:action/actions/AWS_EC2.InstanceId.Reboot/1.0
```

To create an alarm to recover an instance using the CLI

- At a command prompt, type:

```
% aws cloudwatch put-metric-alarm --alarm-name my-Alarm --alarm-description
 "Recover the instance" --namespace "AWS/EC2" --dimensions Name=In
stanceId,Value="i-abc123" --statistic Average  --metric-name StatusCheck
Failed_System --comparison-operator GreaterThanThreshold --threshold 0 --
period 60 --evaluation-periods 2 --alarm-actions arn:aws:automate:us-east-
1:ec2:recover
```

To create an alarm to stop an instance using the API

The following example request shows how to create an alarm that stops an Amazon EC2 instance:

-
```
http://monitoring.amazonaws.com/

?SignatureVersion=2

&Action=PutMetricAlarm

&Version=2009-05-15

&Namespace=AWS/EC2

&MetricName=CPUUtilization

&Dimension.member.1.Name=instance-id

&Dimension.member.1.Value=i-abc123

&Period=86400

&Statistic=Average
```

```
&AlarmName=Stop-EC2-Instance

&ComparisonOperator=LessThanThreshold

&Threshold=10

&EvaluationPeriods=4

&StartTime=2009-01-16T00:00:00

&EndTime=2009-01-16T00:02:00

&Timestamp=2009-01-08-18

&AWSAccessKeyId=XXX YOUR ACCESS KEY XXX

&Signature=%XXX YOUR SIGNATURE XXX%3D

&AlarmActions.member.1=arn:aws:automate:us-east-1:ec2:stop
```

To create an alarm to terminate an instance using the API

The following example request shows how to create an alarm that terminates an Amazon EC2 instance:

- ```
  http://monitoring.amazonaws.com/

  ?SignatureVersion=2

  &Action=PutMetricAlarm

  &Version=2009-05-15

  &Namespace=AWS/EC2

  &MetricName=CPUUtilization

  &Dimension.member.1.Name=instance-id

  &Dimension.member.1.Value=i-abc123

  &Period=86400

  &Statistic=Average

  &AlarmName=Terminate-EC2-Instance

  &ComparisonOperator=LessThanThreshold

  &Threshold=10

  &EvaluationPeriods=4

  &StartTime=2009-01-16T00:00:00

  &EndTime=2009-01-16T00:02:00
  ```
  ```

```
&Timestamp=2009-01-08-18

&AWSAccessKeyId=XXX YOUR ACCESS KEY XXX

&Signature=%XXX YOUR SIGNATURE XXX%3D

&AlarmActions.member.1=arn:aws:automate:us-east-1:ec2:terminate
```

To create an alarm to reboot an instance using the API

The following example request shows how to create an alarm that reboots an Amazon EC2 instance:

- ```
  http://monitoring.amazonaws.com/

  ?SignatureVersion=2

  &Action=PutMetricAlarm

  &Version=2009-05-15

  &Namespace=AWS/EC2

  &MetricName=StatusCheckFailed_Instance

  &Dimension.member.1.Name=instance-id

  &Dimension.member.1.Value=i-abc123

  &Period=60

  &Statistic=Average

  &AlarmName=Reboot-EC2-Instance

  &ComparisonOperator=GreaterThanThreshold

  &Threshold=0

  &EvaluationPeriods=2

  &StartTime=2009-01-16T00:00:00

  &EndTime=2009-01-16T00:02:00

  &Timestamp=2009-01-08-18

  &AWSAccessKeyId=XXX YOUR ACCESS KEY XXX

  &Signature=%XXX YOUR SIGNATURE XXX%3D

  &AlarmActions.member.1=arn:aws:aws:swf:us-east-1:{customer-account}:action/ac
  tions/AWS_EC2.InstanceId.Reboot/1.0
  ```
  ```

To create an alarm to recover an instance using the API

The following example request shows how to create an alarm that recovers an Amazon EC2 instance:

- ```
  http://monitoring.amazonaws.com/

  ?SignatureVersion=2

  &Action=PutMetricAlarm

  &Version=2009-05-15

  &Namespace=AWS/EC2

  &MetricName=StatusCheckFailed_System

  &Dimension.member.1.Name=instance-id

  &Dimension.member.1.Value=i-abc123

  &Period=60

  &Statistic=Average

  &AlarmName=Terminate-EC2-Instance

  &ComparisonOperator=GreaterThanThreshold

  &Threshold=0

  &EvaluationPeriods=2

  &StartTime=2009-01-16T00:00:00

  &EndTime=2009-01-16T00:02:00

  &Timestamp=2009-01-08-18

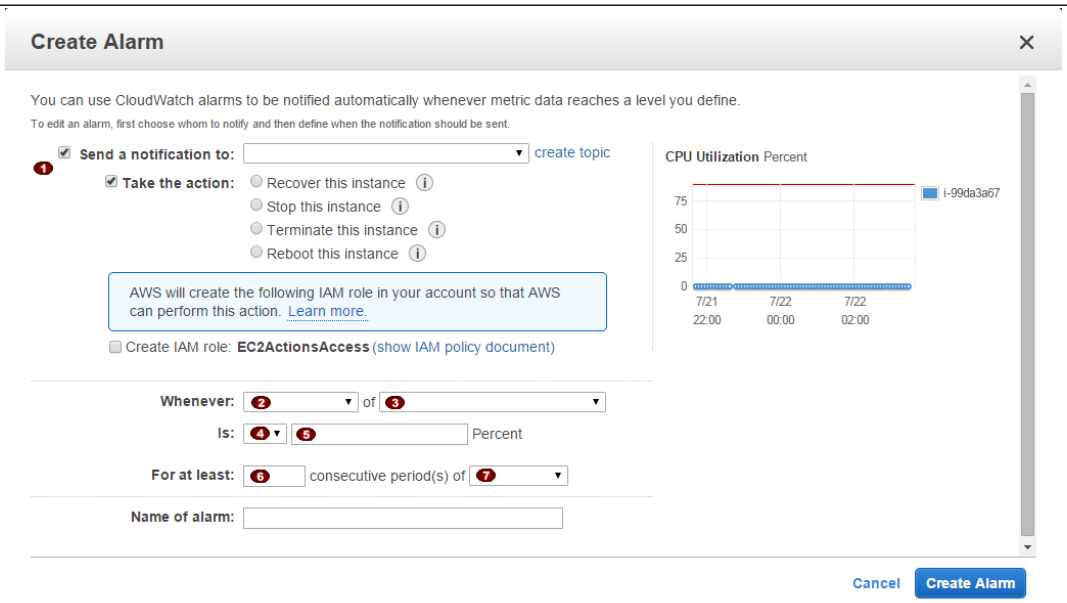
  &AWSAccessKeyId=XXX YOUR ACCESS KEY XXX

  &Signature=%XXX YOUR SIGNATURE XXX%3D

  &AlarmActions.member.1=arn:aws:automate:us-east-1:ec2:recover
  ```

Amazon CloudWatch Alarm Action Scenarios

You can use the Amazon Elastic Compute Cloud (Amazon EC2) console to create alarm actions that stop or terminate an Amazon EC2 instance when certain conditions are met. In the following screen capture of the console page where you set the alarm actions, we've numbered the settings. We've also numbered the settings in the scenarios that follow, to help you create the appropriate actions.
  ```



Scenario 1: Stop Idle Development and Test Instances

Create an alarm that stops an instance used for software development or testing when it has been idle for at least an hour.

Setting	Value
	Stop
	Maximum
	CPUUtilization
	<=
	10%
	60 minutes
	1

Scenario 2: Stop Idle Instances

Create an alarm that stops an instance and sends an email when the instance has been idle for 24 hours.

Setting	Value
	Stop and email
	Average
	CPUUtilization
	<=
	5%
	60 minutes
	24

Scenario 3: Send Email About Web Servers with Unusually High Traffic

Create an alarm that sends email when an instance exceeds 10 GB of outbound network traffic per day.

Setting	Value
	Email
	Sum
	NetworkOut
	>
	10 GB
	1 day
	1

Scenario 4: Stop Web Servers with Unusually High Traffic

Create an alarm that stops an instance and send a text message (SMS) if outbound traffic exceeds 1 GB per hour.

Setting	Value
	Stop and send SMS
	Sum
	NetworkOut
	>
	1 GB
	1 hour
	1

Scenario 5: Stop an Instance Experiencing a Memory Leak

Create an alarm that stops an instance when memory utilization reaches or exceeds 90%, so that application logs can be retrieved for troubleshooting.

Note
The MemoryUtilization metric is a custom metric. In order to use the MemoryUtilization metric, you must install the Perl scripts for Linux instances. For more information, see Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances.

Setting	Value
	Stop
	Maximum
	MemoryUtilization
	>=
	90%
	1 minute
	1

Scenario 6: Stop an Impaired Instance

Create an alarm that stops an instance that fails three consecutive status checks (performed at 5-minute intervals).

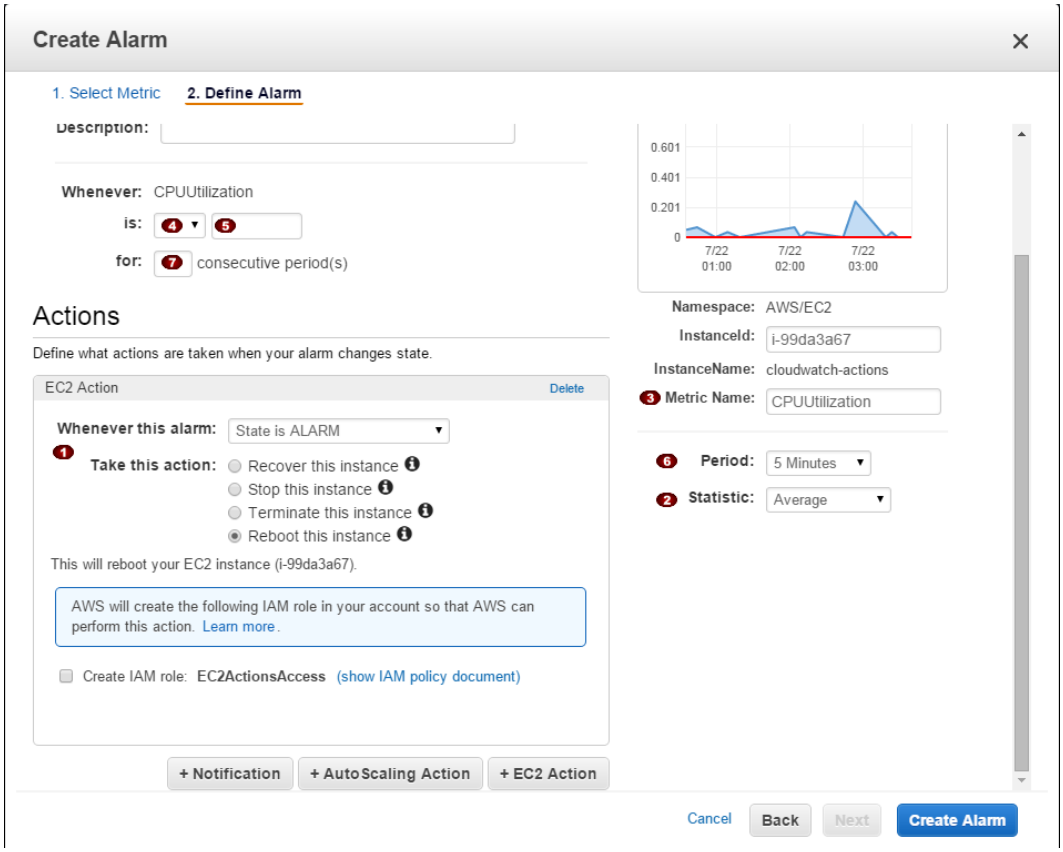
Setting	Value
	Stop
	Average
	StatusCheckFailed_System
	>=
	1
	15 minutes
	1

Scenario 7: Terminate Instances When Batch Processing Jobs Are Complete

Create an alarm that terminates an instance that runs batch jobs when it is no longer sending results data.

Setting	Value
	Terminate
	Maximum
	NetworkOut
	<=
	100,000 bytes
	5 minutes
	1

The previous scenarios can also be performed using the Amazon CloudWatch console. We've numbered the settings on the console to match the numbered settings in the Amazon EC2 console and the scenarios that we covered earlier, so you can make a comparison and create an alarm with the appropriate actions.



Network and Security

Amazon EC2 provides the following network and security features.

Features

- Amazon EC2 Key Pairs and Windows Instances (p. 394)
- Amazon EC2 Security Groups for Windows Instances (p. 398)
- Controlling Access to Amazon EC2 Resources (p. 406)
- Amazon EC2 and Amazon Virtual Private Cloud (p. 449)
- Amazon EC2 Instance IP Addressing (p. 474)
- Elastic IP Addresses (p. 485)
- Elastic Network Interfaces (ENI) (p. 492)
- Placement Groups (p. 504)
- Network Maximum Transmission Unit (MTU) for Your EC2 Instance (p. 507)
- Enabling Enhanced Networking on Windows Instances in a VPC (p. 510)

If you access Amazon EC2 using the command line tools or an API, you'll need your access key ID and secret access key. For more information, see How Do I Get Security Credentials? in the *Amazon Web Services General Reference*.

You can launch an instance into one of two platforms: EC2-Classic or EC2-VPC. An instance that's launched into EC2-Classic or a default VPC is automatically assigned a public IP address. An instance that's launched into a nondefault VPC can be assigned a public IP address on launch. For more information about EC2-Classic and EC2-VPC, see Supported Platforms (p. 455).

Instances can fail or terminate for reasons outside of your control. If an instance fails and you launch a replacement instance, the replacement has a different public IP address than the original. However, if your application needs a static IP address, you can use an *Elastic IP address*.

You can use *security groups* to control who can access your instances. These are analogous to an inbound network firewall that enables you to specify the protocols, ports, and source IP ranges that are allowed to reach your instances. You can create multiple security groups and assign different rules to each group. You can then assign each instance to one or more security groups, and we use the rules to determine which traffic is allowed to reach the instance. You can configure a security group so that only specific IP addresses or specific security groups have access to the instance.

Amazon EC2 Key Pairs and Windows Instances

Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. Public–key cryptography uses a public key to encrypt a piece of data, such as a password, then the recipient uses the private key to decrypt the data. The public and private keys are known as a *key pair*.

To log in to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance. With Windows instances, you use a key pair to obtain the administrator password and then log in using RDP. For more information about key pairs and Linux instances, see Amazon EC2 Key Pairs in the *Amazon EC2 User Guide for Linux Instances*.

Creating a Key Pair

You can use Amazon EC2 to create your key pair. For more information, see Creating Your Key Pair Using Amazon EC2 (p. 394).

Alternatively, you could use a third-party tool and then import the public key to Amazon EC2. For more information, see Importing Your Own Key Pair to Amazon EC2 (p. 395).

Each key pair requires a name. Be sure to choose a name that is easy to remember. Amazon EC2 associates the public key with the name that you specify as the key name.

Amazon EC2 stores the public key only, and you store the private key. Anyone who possesses your private key can decrypt your login information, so it's important that you store your private keys in a secure place.

The keys that Amazon EC2 uses are 2048-bit SSH-2 RSA keys. You can have up to five thousand key pairs per region.

Launching and Connecting to Your Instance

When you launch an instance, you should specify the name of the key pair you plan to use to connect to the instance. If you don't specify the name of an existing key pair when you launch an instance, you won't be able to connect to the instance. When you connect to the instance, you must specify the private key that corresponds to the key pair you specified when you launched the instance. Amazon EC2 doesn't keep a copy of your private key; therefore, if you lose a private key, there is no way to recover it.

Contents

- Creating Your Key Pair Using Amazon EC2 (p. 394)
- Importing Your Own Key Pair to Amazon EC2 (p. 395)
- Retrieving the Public Key for Your Key Pair on Windows (p. 397)
- Verifying Your Key Pair's Fingerprint (p. 397)
- Deleting Your Key Pair (p. 398)

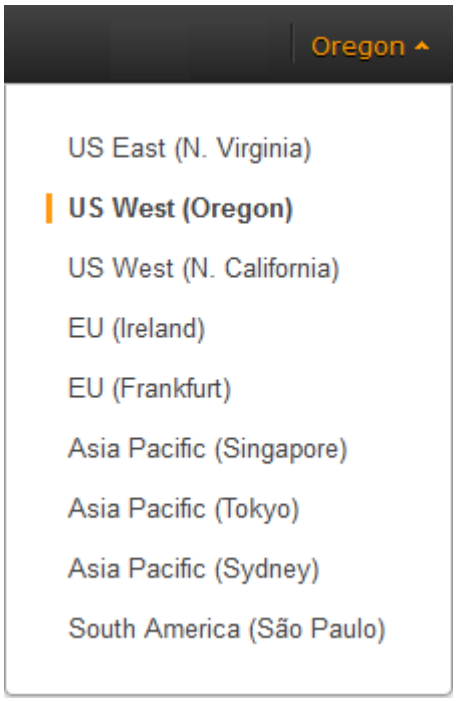
Creating Your Key Pair Using Amazon EC2

You can create a key pair using the Amazon EC2 console or the command line. After you create a key pair, you can specify it when you launch your instance.

To create your key pair using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select a region for the key pair. You can select any region that's available to you, regardless of your location. This choice is important because some Amazon EC2 resources

can be shared between regions, but key pairs can't. For example, if you create a key pair in the US West (Oregon) region, you can't see or use the key pair in another region.



3. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
4. Choose **Create Key Pair**.
5. Enter a name for the new key pair in the **Key pair name** field of the **Create Key Pair** dialog box, and then choose **Create**.
6. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is `.pem`. Save the private key file in a safe place.

   Important
   This is the only chance for you to save the private key file. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

To create your key pair using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- create-key-pair (AWS CLI)
- ec2-create-keypair (Amazon EC2 CLI)
- New-EC2KeyPair (AWS Tools for Windows PowerShell)

Importing Your Own Key Pair to Amazon EC2

If you used Amazon EC2 to create your key pair, as described in the previous section, you are ready to launch an instance. Otherwise, instead of using Amazon EC2 to create your key pair, you can create an RSA key pair using a third-party tool and then import the public key to Amazon EC2. For example, you can use **ssh-keygen** (a tool provided with the standard OpenSSH installation) to create a key pair.

Alternatively, Java, Ruby, Python, and many other programming languages provide standard libraries that you can use to create an RSA key pair.

Amazon EC2 accepts the following formats:

- OpenSSH public key format
- Base64 encoded DER format
- SSH public key file format as specified in RFC4716

Amazon EC2 does not accept DSA keys. Make sure your key generator is set up to create RSA keys.

Supported lengths: 1024, 2048, and 4096.

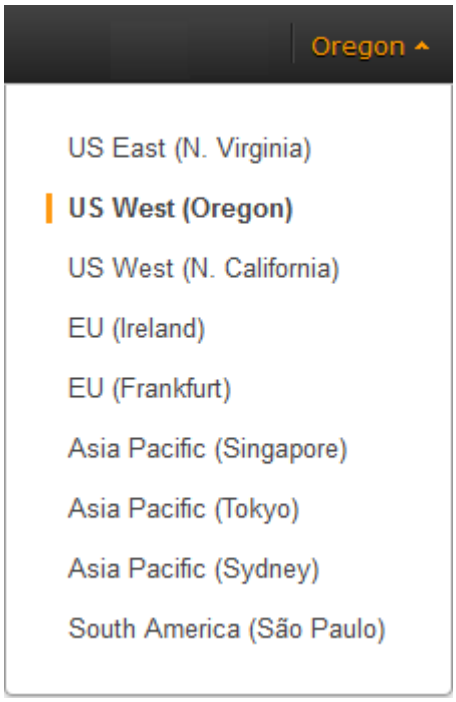
To create a key pair using a third-party tool

1. Generate a key pair with a third-party tool of your choice.
2. Save the public key to a local file. For example, `C:\keys\my-key-pair.pub`. The file name extension for this file is not important.
3. Save the private key to a different local file that has the `.pem` extension. For example, `C:\keys\my-key-pair.pem`. Save the private key file in a safe place. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

Use the following steps to import your key pair using the Amazon EC2 console. (If you prefer, you can use the ec2-import-keypair command or the ImportKeyPair action to import the public key.)

To import the public key

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the region for the key pair. This choice is important because key pair resources cannot be shared between regions. For example, if you import a key pair into the US West (Oregon) region, you won't be able to see or use the key pair in another region.



3. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
4. Choose **Import Key Pair**.
5. In the **Import Key Pair** dialog box, choose **Browse**, and select the public key file that you saved previously. Enter a name for the key pair in the **Key pair name** field, and choose **Import**.

After the public key file is imported, you can verify that the key pair was imported successfully using the Amazon EC2 console as follows. (If you prefer, you can use the ec2-describe-keypairs command or the DescribeKeyPairs action to list your key pairs.)

To verify that your key pair was imported

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the region in which you created the key pair.
3. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
4. Verify that the key pair that you imported is in the displayed list of key pairs.

Retrieving the Public Key for Your Key Pair on Windows

On Windows, you can use PuTTYgen to get the public key for your key pair. Start PuTTYgen, click **Load**, and select the `.ppk` or `.pem` file. PuTTYgen displays the public key.

The public key that you specified when you launched an instance is also available to you through its instance metadata. To view the public key that you specified when launching the instance, use the following command from your instance:

```
C:\> GET http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

For more information, see Retrieving Instance Metadata (p. 161).

Verifying Your Key Pair's Fingerprint

On the **Key Pairs** page in the Amazon EC2 console, the **Fingerprint** column displays the fingerprints generated from your key pairs. AWS calculates the fingerprint differently depending on whether the key pair was generated by AWS or a third-party tool. If you created the key pair using AWS, the fingerprint is calculated using an SHA-1 hash function. If you created the key pair with a third-party tool and uploaded the public key to AWS, or if you generated a new public key from an existing AWS-created private key and uploaded it to AWS, the fingerprint is calculated using an MD5 hash function.

You can use the fingerprint that's displayed on the **Key Pairs** page to verify that the private key you have on your local machine matches the public key that's stored in AWS.

If you created your key pair using AWS, you can use the ec2-fingerprint-key command in the Amazon EC2 CLI to generate a fingerprint from the private key file on your local machine. The output should match the fingerprint that's displayed in the console. Alternatively, you can use the OpenSSL tools to generate a fingerprint from the private key file:

```
C:\> openssl pkcs8 -in path_to_private_key -inform PEM -outform DER -topk8 -
nocrypt | openssl sha1 -c
```

If you created your key pair using a third-party tool and uploaded the public key to AWS, you can use the OpenSSL tools to generate a fingerprint from the private key file on your local machine:

```
C:\> openssl rsa -in path_to_private_key -pubout -outform DER | openssl md5 -c
```

The output should match the fingerprint that's displayed in the console.

Deleting Your Key Pair

When you delete a key pair, you are only deleting Amazon EC2's copy of the public key. Deleting a key pair doesn't affect the private key on your computer or the public key on any instances already launched using that key pair. You can't launch a new instance using a deleted key pair, but you can continue to connect to any instances that you launched using a deleted key pair, as long as you still have the private key (`.pem`) file.

You can delete a key pair using the Amazon EC2 console or the command line.

To delete your key pair using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
3. Select the key pair and choose **Delete**.
4. When prompted, choose **Yes**.

To delete your key pair using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- delete-key-pair (AWS CLI)
- ec2-delete-keypair (Amazon EC2 CLI)
- Remove-EC2KeyPair (AWS Tools for Windows PowerShell)

Amazon EC2 Security Groups for Windows Instances

A *security group* acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group. When we decide whether to allow traffic to reach an instance, we evaluate all the rules from all the security groups that are associated with the instance.

If you need to allow traffic to a Linux instance, see Amazon EC2 Security Groups for Linux Instances in the *Amazon EC2 User Guide for Linux Instances*.

Topics

- Security Groups for EC2-Classic (p. 399)

- Security Groups for EC2-VPC (p. 399)
- Security Group Rules (p. 400)
- Default Security Groups (p. 401)
- Custom Security Groups (p. 401)
- Creating a Security Group (p. 402)
- Describing Your Security Groups (p. 403)
- Adding Rules to a Security Group (p. 403)
- Deleting Rules from a Security Group (p. 404)
- Deleting a Security Group (p. 405)
- API and Command Overview (p. 405)

If you have requirements that aren't met by security groups, you can maintain your own firewall on any of your instances in addition to using security groups.

Security Groups for EC2-Classic

If you're using EC2-Classic, you must use security groups created specifically for EC2-Classic. When you launch an instance in EC2-Classic, you must specify a security group in the same region as the instance. You can't specify a security group that you created for a VPC when you launch an instance in EC2-Classic.

After you launch an instance in EC2-Classic, you can't change its security groups. However, you can add rules to or remove rules from a security group, and those changes are automatically applied to all instances that are associated with the security group.

Note
In EC2-Classic, you can associate an instance with up to 500 security groups and add up to 100 rules to a security group.

Security Groups for EC2-VPC

If you're using EC2-VPC, you must use security groups created specifically for your VPC. When you launch an instance in a VPC, you must specify a security group for that VPC. You can't specify a security group that you created for EC2-Classic when you launch an instance in a VPC.

After you launch an instance in a VPC, you can change its security groups. Security groups are associated with network interfaces. Changing an instance's security groups changes the security groups associated with the primary network interface (eth0). For more information, see Changing an Instance's Security Groups in the *Amazon VPC User Guide*. You can also change the security groups associated with any other network interface. For more information, see Changing the Security Group of an Elastic Network Interface (p. 500).

You can change the rules of a security group, and those changes are automatically applied to all instances that are associated with the security group.

Note
In EC2-VPC, you can associate a network interface with up to 5 security groups and add up to 50 rules to a security group.

When you specify a security group for a nondefault VPC to the CLI or the API actions, you must use the security group ID and not the security group name to identify the security group.

Security groups for EC2-VPC have additional capabilities that aren't supported by security groups for EC2-Classic. For more information about security groups for EC2-VPC, see Security Groups for Your VPC in the *Amazon VPC User Guide*.

Security Group Rules

The rules of a security group control the inbound traffic that's allowed to reach the instances that are associated with the security group and the outbound traffic that's allowed to leave them. By default, security groups allow all outbound traffic.

You can add and remove rules at any time. Your changes are automatically applied to the instances associated with the security group after a short period. You can either edit an existing rule in a security group, or delete it and add a new rule. You can copy the rules from an existing security group to a new security group. You can't change the outbound rules for EC2-Classic. Security group rules are always permissive; you can't create rules that deny access.

For each rule, you specify the following:

- The protocol to allow (such as TCP, UDP, or ICMP).
- TCP and UDP, or a custom protocol: The range of ports to allow
- ICMP: The ICMP type and code
- One or the following options for the source (inbound rules) or destination (outbound rules):
  - An individual IP address, in CIDR notation. Be sure to use the `/32` prefix after the IP address; if you use the `/0` prefix after the IP address, this opens the port to everyone. For example, specify the IP address `203.0.113.1` as `203.0.113.1/32`.
  - An IP address range, in CIDR notation (for example, `203.0.113.0/24`).
  - The name (EC2-Classic) or ID (EC2-Classic or EC2-VPC) of a security group. This allows instances associated with the specified security group to access instances associated with this security group. (Note that this does not add rules from the source security group to this security group.) You can specify one of the following security groups:
    - The current security group.
    - EC2-Classic: A different security group for EC2-Classic in the same region
    - EC2-VPC: A different security group for the same VPC
    - EC2-Classic: A security group for another AWS account in the same region (add the AWS account ID as a prefix; for example, `111122223333/sg-edcd9784`)

When you specify a security group as the source or destination for a rule, the rule affects all instances associated with the security group. Incoming traffic is allowed based on the private IP addresses of the instances that are associated with the source security group (and not the public IP or Elastic IP addresses).

If there is more than one rule for a specific port, we apply the most permissive rule. For example, if you have a rule that allows access to TCP port 3389 (RDP) from IP address `203.0.113.1` and another rule that allows access to TCP port 3389 from everyone, everyone has access to TCP port 3389.

When you associate multiple security groups with an instance, the rules from each security group are effectively aggregated to create one set of rules. We use this set of rules to determine whether to allow access.

Caution
Because you can assign multiple security groups to an instance, an instance can have hundreds of rules that apply. This might cause problems when you access the instance. Therefore, we recommend that you condense your rules as much as possible.

For more information about IP addresses, see Amazon EC2 Instance IP Addressing (p. 474). For more information about creating security group rules to ensure that Path MTU Discovery can function correctly, see Path MTU Discovery (p. 508).

Default Security Groups

Your AWS account automatically has a *default security group* per region for EC2-Classic. When you create a VPC, we automatically create a default security group for the VPC. If you don't specify a different security group when you launch an instance, the instance is automatically associated with the appropriate default security group.

A default security group is named `default`, and it has an ID assigned by AWS. The following are the initial settings for each default security group:

- Allow inbound traffic only from other instances associated with the default security group
- Allow all outbound traffic from the instance

The default security group specifies itself as a source security group in its inbound rules. This is what allows instances associated with the default security group to communicate with other instances associated with the default security group.

You can change the rules for a default security group. For example, you can add an inbound rule to allow RDP connections so that specific hosts can manage the instance.

You can't delete a default security group. If you try to delete the EC2-Classic default security group, you'll get the following error: `Client.InvalidGroup.Reserved: The security group 'default' is reserved`. If you try to delete a VPC default security group, you'll get the following error: `Client.CannotDelete: the specified group: "sg-51530134" name: "default" cannot be deleted by a user`.

Custom Security Groups

If you don't want all your instances to use the default security group, you can create your own security groups and specify them when you launch your instances. You can create multiple security groups to reflect the different roles that your instances play; for example, a web server or a database server. For instructions that help you create security groups for web servers and database servers, see Recommended Security Groups in the *Amazon VPC User Guide*.

> **Note**
> In EC2-Classic, you can create up to 500 security groups in each region for each account. In EC2-VPC, you can create up to 100 security groups per VPC. The security groups for EC2-Classic do not count against the security group limit for EC2-VPC.

When you create a security group, you must provide it with a name and a description. Security group names and descriptions can be up to 255 characters in length, and are limited to the following characters:

- EC2-Classic: ASCII characters
- EC2-VPC: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

AWS assigns each security group a unique ID in the form sg-*xxxxxxxx*. The following are the initial settings for a security group that you create:

- Allow no inbound traffic
- Allow all outbound traffic

After you've created a security group, you can change its inbound rules to reflect the type of inbound traffic that you want to reach the associated instances. In EC2-VPC, you can also change its outbound rules.

To allow instances that have the same security group to communicate with each other, you must explicitly add rules for this. The following table describes the rules that you must add to your security group to enable instances in EC2-Classic to communicate with each other.

Inbound			
Source	**Protocol**	**Port Range**	**Comments**
The ID of the security group	ICMP	All	Allow inbound ICMP access from other instances associated with this security group
The ID of the security group	TCP	0 - 65535	Allow inbound TCP access from other instances associated with this security group
The ID of the security group	UDP	0 - 65535	Allow inbound UDP access from other instances associated with this security group

The following table describes the rules that you must add to your security group to enable instances in a VPC to communicate with each other.

Inbound			
Source	**Protocol**	**Port Range**	**Comments**
The ID of the security group	All	All	Allow inbound traffic from other instances associated with this security group

Creating a Security Group

To create a new security group

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Security Groups**.
3. Click **Create Security Group**.
4. Specify a name and description for the security group. For **VPC**, select **No VPC** to create a security group for EC2-Classic, or select a VPC ID to create a security group for that VPC.
5. You can start adding rules, or you can click **Create** to create the security group now (you can always add rules later). For more information about adding rules, see Adding Rules to a Security Group (p. 403).

To copy a security group

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Security Groups**.
3. Select the security group you want to copy, click **Actions**, and then select **Copy to new**.
4. The **Create Security Group** dialog opens, and is populated with the rules from the existing security group. Specify a name and description for your new security group. In the **VPC** list, select **No VPC** to create a security group for EC2-Classic, or select a VPC ID to create a security group for that VPC. When you are done, click **Create**.

You can assign a security group to an instance when you launch the instance. When you add or remove rules, those changes are automatically applied to all instances to which you've assigned the security group.

After you launch an instance in EC2-Classic, you can't change its security groups. After you launch an instance in a VPC, you can change its security groups. For more information, see Changing an Instance's Security Groups in the *Amazon VPC User Guide*.

Describing Your Security Groups

To describe your security groups for EC2-Classic

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Security Groups**.
3. Select **Network Platforms** from the filter list, then select **EC2-Classic**.
4. Select a security group. We display general information in the **Description** tab and inbound rules on the **Inbound** tab.

To describe your security groups for EC2-VPC

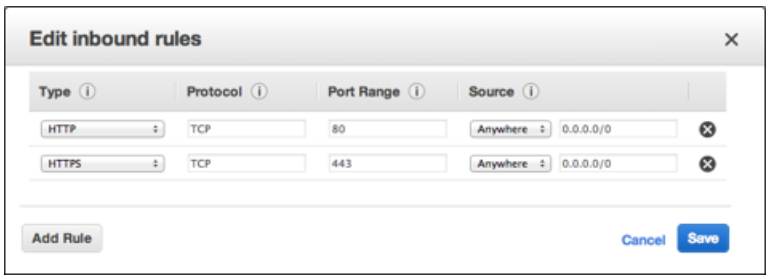
1. Open the Amazon EC2 console.
2. In the navigation pane, click **Security Groups**.
3. Select **Network Platforms** from the filter list, then select **EC2-VPC**.
4. Select a security group. We display general information in the **Description** tab, inbound rules on the **Inbound** tab, and outbound rules on the **Outbound** tab.

Adding Rules to a Security Group

When you add a rule to a security group, the new rule is automatically applied to any instances associated with the security group.

To add rules to a security group

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Security Groups**.
3. Select the security group.
4. You can allow web servers to receive all inbound HTTP and HTTPS traffic. On the **Inbound** tab, click **Edit**. In the dialog, click **Add Rule**. Select **HTTP** from the **Type** list, and leave the source as **Anywhere** (`0.0.0.0/0`). Add a similar rule for the HTTPS protocol.

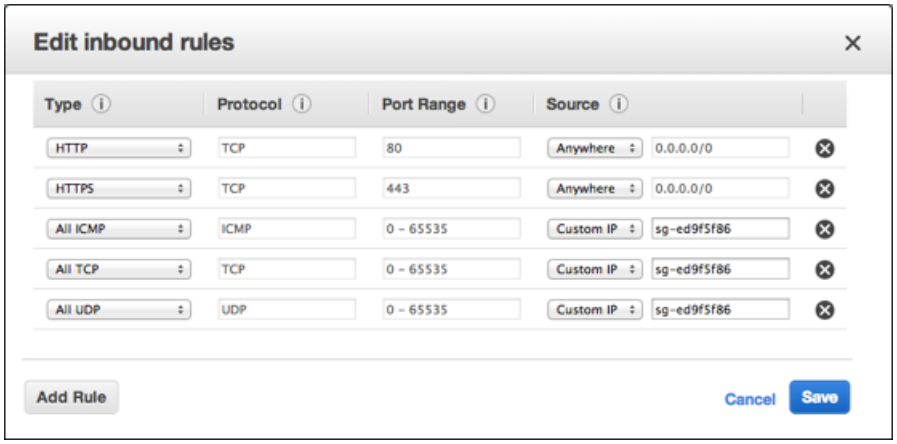


5. To connect to a Windows instance, you need to allow RDP traffic. Click **Add Rule**, and then select **RDP** from the **Type** list.

In the **Source** field, specify the public IP address of your computer, in CIDR notation. For example, if your IP address is `203.0.113.25`, specify `203.0.113.25/32` to list this single IP address in CIDR notation. If your company allocates addresses from a range, specify the entire range, such as `203.0.113.0/24`. You can select **My IP** to from the **Source** list to let us automatically populate the field with your computer's IP address. However, if you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

Caution
If you use `0.0.0.0/0`, you enable all IP addresses to access your instance using RDP. This is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, you'll authorize only a specific IP address or range of addresses to access your instance.

6. You can allow communication between all instances associated with this security group, or between instances associated with another security group and instances associated with this security group. Click **Add Rule**, select **All ICMP**, then start typing the ID of the security group in **Source**; this provides you with a list of security groups. Select the security group from the list. Repeat the steps for the TCP and UDP protocols. Click **Save** when you are done.



7. If you are creating a security group for a VPC, you can also specify outbound rules. For an example, see Adding and Removing Rules in the *Amazon VPC User Guide*.

Deleting Rules from a Security Group

When you delete a rule from a security group, the change is automatically applied to any instances associated with the security group.

To delete a security group rule

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Security Groups**.
3. Select a security group.
4. Click **Edit**, and then click the **Delete** icon next to each rule that you need to delete.
5. Click **Save**.

Deleting a Security Group

You can't delete a security group that is associated with an instance. You can't delete the default security group. You can't delete a security group that is referenced by a rule in another security group. If your security group is referenced by one of its own rules, you must delete the rule before you can delete the security group.

To delete a security group

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Security Groups**.
3. Select a security group and click **Delete**.
4. Click **Yes, Delete**.

API and Command Overview

You can perform the tasks described on this page using the command line or an API. For more information about the command line interfaces and a list of available APIs, see Accessing Amazon EC2 (p. 3).

Create a security group

- create-security-group (AWS CLI)
- ec2-create-group (Amazon EC2 CLI)
- New-EC2SecurityGroup (AWS Tools for Windows PowerShell)

Add one or more ingress rules to a security group

- authorize-security-group-ingress (AWS CLI)
- ec2-authorize (Amazon EC2 CLI)
- Grant-EC2SecurityGroupIngress (AWS Tools for Windows PowerShell)

[EC2-VPC] Add one or more egress rules to a security group

- authorize-security-group-egress (AWS CLI)
- ec2-authorize (Amazon EC2 CLI)
- Grant-EC2SecurityGroupIngress (AWS Tools for Windows PowerShell)

Describe one or more security groups

- describe-security-groups (AWS CLI)
- ec2-describe-group (Amazon EC2 CLI)
- Get-EC2SecurityGroup (AWS Tools for Windows PowerShell)

[EC2-VPC] Modify the security groups for an instance

- modify-instance-attribute (AWS CLI)
- ec2-modify-instance-attribute (Amazon EC2 CLI)
- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

Remove one or more ingress rules from a security group

- revoke-security-group-ingress (AWS CLI)
- ec2-revoke (Amazon EC2 CLI)
- Revoke-EC2SecurityGroupIngress (AWS Tools for Windows PowerShell)

[EC2-VPC] Remove one or more egress rules from a security group

- revoke-security-group-egress(AWS CLI)
- ec2-revoke (Amazon EC2 CLI)
- Revoke-EC2SecurityGroupEgress (AWS Tools for Windows PowerShell)

Delete a security group

- delete-security-group (AWS CLI)
- ec2-delete-group (Amazon EC2 CLI)
- Remove-EC2SecurityGroup (AWS Tools for Windows PowerShell)

Controlling Access to Amazon EC2 Resources

Your security credentials identify you to services in AWS and grant you unlimited use of your AWS resources, such as your Amazon EC2 resources. You can use features of Amazon EC2 and AWS Identity and Access Management (IAM) to allow other users, services, and applications to use your Amazon EC2 resources without sharing your security credentials. You can use IAM to control how other users use resources in your AWS account, and you can use security groups to control access to your Amazon EC2 instances. You can choose to allow full use or limited use of your Amazon EC2 resources.

Contents

- Network Access to Your Instance (p. 406)
- Amazon EC2 Permission Attributes (p. 406)
- IAM and Amazon EC2 (p. 407)
- IAM Policies for Amazon EC2 (p. 408)
- IAM Roles for Amazon EC2 (p. 442)
- Authorizing Inbound Traffic for Your Windows Instances (p. 448)

Network Access to Your Instance

A security group acts as a firewall that controls the traffic allowed to reach one or more instances. When you launch an instance, you assign it one or more security groups. You add rules to each security group that control traffic for the instance. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances to which the security group is assigned.

For more information, see Authorizing Inbound Traffic for Your Windows Instances (p. 448).

Amazon EC2 Permission Attributes

Your organization might have multiple AWS accounts. Amazon EC2 enables you to specify additional AWS accounts that can use your Amazon Machine Images (AMIs) and Amazon EBS snapshots. These permissions work at the AWS account level only; you can't restrict permissions for specific users within

the specified AWS account. All users in the AWS account that you've specified can use the AMI or snapshot.

Each AMI has a `LaunchPermission` attribute that controls which AWS accounts can access the AMI. For more information, see Making an AMI Public (p. 60).

Each Amazon EBS snapshot has a `createVolumePermission` attribute that controls which AWS accounts can use the snapshot. For more information, see Sharing an Amazon EBS Snapshot (p. 554).

IAM and Amazon EC2

IAM enables you to do the following:

- Create users and groups under your AWS account
- Assign unique security credentials to each user under your AWS account
- Control each user's permissions to perform tasks using AWS resources
- Allow the users in another AWS account to share your AWS resources
- Create roles for your AWS account and define the users or services that can assume them
- Use existing identities for your enterprise to grant permissions to perform tasks using AWS resources

By using IAM with Amazon EC2, you can control whether users in your organization can perform a task using specific Amazon EC2 API actions and whether they can use specific AWS resources.

This topic helps you answer the following questions:

- How do I create groups and users in IAM?
- How do I create a policy?
- What IAM policies do I need to carry out tasks in Amazon EC2?
- How do I grant permissions to perform actions in Amazon EC2?
- How do I grant permissions to perform actions on specific resources in Amazon EC2?

Creating an IAM Group and Users

To create an IAM group

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, click **Groups** and then click **Create New Group**.
3. In the **Group Name** box, type a name for your group, and then click **Next Step**.
4. On the **Attach Policy** page, select an AWS managed policy. For example, for Amazon EC2, one of the following AWS managed policies might meet your needs:

   - PowerUserAccess
   - ReadOnlyAccess
   - AmazonEC2FullAccess
   - AmazonEC2ReadOnlyAccess

5. Click **Next Step** and then click **Create Group**.

Your new group is listed under **Group Name**.

To create an IAM user, add the user to your group, and create a password for the user

1. In the navigation pane, click **Users** and then click **Create New Users**.
2. In box **1**, type a user name and then click **Create**.
3. Click **Download Credentials** and save your access key in a secure place. You will need your access key for programmatic access to AWS using the AWS CLI, the AWS SDKs, or the HTTP APIs.

   Note
   You cannot retrieve the secret access key after you complete this step; if you misplace it you must create a new one.

   After you have downloaded your access key, click **Close**.
4. Under **User Name**, click the name of the user you just created.
5. Click **Groups** and then click **Add User to Groups**.
6. Select the group you created earlier, and then click **Add to Groups**.
7. Click **Security Credentials** and then under **Sign-In Credentials**, click **Manage Password**.
8. Select **Assign a custom password** and then type and confirm a password. When you are finished, click **Apply**.
9. Give each user his or her credentials (access keys and password); this enables them to use services based on the permissions you specified for the IAM group

Related Topics

For more information about IAM, see the following:

- IAM Policies for Amazon EC2 (p. 408)
- IAM Roles for Amazon EC2 (p. 442)
- Identity and Access Management (IAM)
- IAM User Guide

IAM Policies for Amazon EC2

By default, IAM users don't have permission to create or modify Amazon EC2 resources, or perform tasks using the Amazon EC2 API. (This means that they also can't do so using the Amazon EC2 console or CLI.) To allow IAM users to create or modify resources and perform tasks, you must create IAM policies that grant IAM users permission to use the specific resources and API actions they'll need, and then attach those policies to the IAM users or groups that require those permissions.

When you attach a policy to a user or group of users, it allows or denies the users permission to perform the specified tasks on the specified resources. For more general information about IAM policies, see Permissions and Policies in the *IAM User Guide*. For more information about managing and creating custom IAM policies, see Managing IAM Policies.

Getting Started

An IAM policy must grant or deny permission to use one or more Amazon EC2 actions. It must also specify the resources that can be used with the action, which can be all resources, or in some cases, specific resources. The policy can also include conditions that you apply to the resource.

Amazon EC2 partially supports resource-level permissions. This means that for some EC2 API actions, you cannot specify which resource a user is allowed to work with for that action; instead, you have to allow users to work with all resources for that action.

Task	Topic
Understand the basic structure of a policy	Policy Syntax (p. 409)
Define actions in your policy	Actions for Amazon EC2 (p. 410)
Define specific resources in your policy	Amazon Resource Names for Amazon EC2 (p. 410)
Apply conditions to the use of the resources	Condition Keys for Amazon EC2 (p. 413)
Work with the available resource-level permissions for Amazon EC2	Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 416)
Test your policy	Checking that Users Have the Required Permissions (p. 415)
Example policies for a CLI or SDK	Example Policies for Working With the AWS CLI, the Amazon EC2 CLI, or an AWS SDK (p. 424)
Example policies for the Amazon EC2 console	Example Policies for Working in the Amazon EC2 Console (p. 434)

Policy Structure

The following topics explain the structure of an IAM policy.

Topics

- Policy Syntax (p. 409)
- Actions for Amazon EC2 (p. 410)
- Amazon Resource Names for Amazon EC2 (p. 410)
- Condition Keys for Amazon EC2 (p. 413)
- Checking that Users Have the Required Permissions (p. 415)

Policy Syntax

An IAM policy is a JSON document that consists of one of more statements. Each statement is structured as follows:

```
{
  "Statement":[{
    "Effect":"effect",
    "Action":"action",
    "Resource":"arn",
    "Condition":{
      "condition":{
        "key":"value"
        }
      }
    }
  ]
}
```

There are various elements that make up a statement:

- **Effect:** The *effect* can be `Allow` or `Deny`. By default, IAM users don't have permission to use resources and API actions, so all requests are denied. An explicit allow overrides the default. An explicit deny overrides any allows.
- **Action**: The *action* is the specific API action for which you are granting or denying permission. To learn about specifying *action*, see Actions for Amazon EC2 (p. 410).
- **Resource**: The resource that's affected by the action. Some Amazon EC2 API actions allow you to include specific resources in your policy that can be created or modified by the action. To specify a resource in the statement, you need to use its Amazon Resource Name (ARN). For more information about specifying the *arn* value, see Amazon Resource Names for Amazon EC2 (p. 410). For more information about which API actions support which ARNs, see Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 416). If the API action does not support ARNs, use the * wildcard to specify that all resources can be affected by the action.
- **Condition**: Conditions are optional. They can be used to control when your policy will be in effect. For more information about specifying conditions for Amazon EC2, see Condition Keys for Amazon EC2 (p. 413).

For more information about example IAM policy statements for Amazon EC2, see Example Policies for Working With the AWS CLI, the Amazon EC2 CLI, or an AWS SDK (p. 424).

Actions for Amazon EC2

In an IAM policy statement, you can specify any API action from any service that supports IAM. For Amazon EC2, use the following prefix with the name of the API action: `ec2:`. For example: `ec2:RunInstances` and `ec2:CreateImage`.

To specify multiple actions in a single statement, separate them with commas as follows:

```
"Action": ["ec2:action1", "ec2:action2"]
```

You can also specify multiple actions using wildcards. For example, you can specify all actions whose name begins with the word "Describe" as follows:

```
"Action": "ec2:Describe*"
```

To specify all Amazon EC2 API actions, use the * wildcard as follows:

```
"Action": "ec2:*"
```

For a list of Amazon EC2 actions, see Actions in the *Amazon EC2 API Reference*.

Amazon Resource Names for Amazon EC2

Each IAM policy statement applies to the resources that you specify using their ARNs.

Important
Currently, not all API actions support individual ARNs; we'll add support for additional API actions and ARNs for additional Amazon EC2 resources later. For information about which ARNs you can use with which Amazon EC2 API actions, as well as supported condition keys for each ARN, see Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 416).

An ARN has the following general syntax:

```
arn:aws:[service]:[region]:[account]:resourceType/resourcePath
```

service
The service (for example, `ec2`).

region
The region for the resource (for example, `us-east-1`).

account
The AWS account ID, with no hyphens (for example, `123456789012`).

resourceType
The type of resource (for example, `instance`).

resourcePath
A path that identifies the resource. You can use the * wildcard in your paths.

For example, you can indicate a specific instance (`i-1a2b3c4d`) in your statement using its ARN as follows:

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/i-1a2b3c4d"
```

You can also specify all instances that belong to a specific account by using the * wildcard as follows:

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*"
```

To specify all resources, or if a specific API action does not support ARNs, use the * wildcard in the `Resource` element as follows:

```
"Resource": "*"
```

The following table describes the ARNs for each type of resource used by the Amazon EC2 API actions.

Resource Type	ARN
All Amazon EC2 resources	arn:aws:ec2:*
All Amazon EC2 resources owned by the specified account in the specified region	arn:aws:ec2:*region*:*account*:*
Customer gateway	arn:aws:ec2:*region*:*account*:customer-gateway/*cgw-id* Where *cgw-id* is cgw-*xxxxxxxx*
DHCP options set	arn:aws:ec2:*region*:*account*:dhcp-options/*dhcp-options-id* Where *dhcp-options-id* is dopt-*xxxxxxxx*
Image	arn:aws:ec2:*region*::image/*image-id* Where *image-id* is the ID of the AMI, AKI, or ARI, and *account* isn't used
Instance	arn:aws:ec2:*region*:*account*:instance/*instance-id* Where *instance-id* is i-*xxxxxxxx*

Resource Type	ARN
Instance profile	arn:aws:iam::*account*:instance-profile/*instance-profile-name* Where *instance-profile-name* is the name of the instance profile, and *region* isn't used
Internet gateway	arn:aws:ec2:*region*:*account*:internet-gateway/*igw-id* Where *igw-id* is igw-*xxxxxxxx*
Key pair	arn:aws:ec2:*region*:*account*:key-pair/*key-pair-name* Where *key-pair-name* is the key pair name (for example, `gsg-keypair`)
Network ACL	arn:aws:ec2:*region*:*account*:network-acl/*nacl-id* Where *nacl-id* is acl-*xxxxxxxx*
Network interface	arn:aws:ec2:*region*:*account*:network-interface/*eni-id* Where *eni-id* is eni-*xxxxxxxx*
Placement group	arn:aws:ec2:*region*:*account*:placement-group/*placement-group-name* Where *placement-group-name* is the placement group name (for example, `my-cluster`)
Route table	arn:aws:ec2:*region*:*account*:route-table/*route-table-id* Where *route-table-id* is rtb-*xxxxxxxx*
Security group	arn:aws:ec2:*region*:*account*:security-group/*security-group-id* Where *security-group-id* is sg-*xxxxxxxx*
Snapshot	arn:aws:ec2:*region*::snapshot/*snapshot-id* Where *snapshot-id* is snap-*xxxxxxxx*, and *account* isn't used
Subnet	arn:aws:ec2:*region*:*account*:subnet/*subnet-id* Where *subnet-id* is subnet-*xxxxxxxx*
Volume	arn:aws:ec2:*region*:*account*:volume/*volume-id* Where *volume-id* is vol-*xxxxxxxx*
VPC	arn:aws:ec2:*region*:*account*:vpc/*vpc-id* Where *vpc-id* is vpc-*xxxxxxxx*
VPC peering connection	arn:aws:ec2:*region*:*account*:vpc-peering-connection/*vpc-peering-connection-id* Where *vpc-peering connection-id* is pcx-*xxxxxxxx*

Many Amazon EC2 API actions involve multiple resources. For example, `AttachVolume` attaches an Amazon EBS volume to an instance, so an IAM user must have permission to use the volume and the instance. To specify multiple resources in a single statement, separate their ARNs with commas, as follows:

```
"Resource": ["arn1", "arn2"]
```

For more general information about ARNs, see Amazon Resource Names (ARN) and AWS Service Namespaces in the *Amazon Web Services General Reference*. For more information about the resources that are created or modified by the Amazon EC2 actions, and the ARNs that you can use in your IAM policy statements, see Granting IAM Users Required Permissions for Amazon EC2 Resources in the *Amazon EC2 API Reference*.

Condition Keys for Amazon EC2

In a policy statement, you can optionally specify conditions that control when it is in effect. Each condition contains one or more key-value pairs. Condition keys are not case sensitive. We've defined AWS-wide condition keys, plus additional service-specific condition keys.

If you specify multiple conditions, or multiple keys in a single condition, we evaluate them using a logical AND operation. If you specify a single condition with multiple values for one key, we evaluate the condition using a logical OR operation. For permission to be granted, all conditions must be met.

You can also use placeholders when you specify conditions. For example, you can grant an IAM user permission to use resources with a tag that specifies his or her IAM user name. For more information, see Policy Variables in the *IAM User Guide*.

Amazon EC2 implements the AWS-wide condition keys (see Available Keys), plus the following service-specific condition keys. (We'll add support for additional service-specific condition keys for Amazon EC2 later.)

Condition Key	Key/Value Pair	Evaluation Types
ec2:AccepterVpc	"ec2:AccepterVpc":"*vpc-arn*" Where *vpc-arn* is the VPC ARN for the peer VPC	ARN, Null
ec2:AvailabilityZone	"ec2:AvailabilityZone":"*az-api-name*" Where *az-api-name* is the name of the Availability Zone (for example, `us-west-2a`) To list your Availability Zones, use ec2-describe-availability-zones	String, Null
ec2:EbsOptimized	"ec2:EbsOptimized":"*optimized-flag*" Where *optimized-flag* is `true` | `false`	Boolean, Null
ec2:ImageType	"ec2:ImageType":"*image-type-api-name*" Where *image-type-api-name* is `ami` | `aki` | `ari`	String, Null
ec2:InstanceProfile	"ec2:InstanceProfile":"*instance-profile-arn*" Where *instance-profile-arn* is the instance profile ARN	ARN, Null

Condition Key	Key/Value Pair	Evaluation Types
ec2:InstanceType	"ec2:InstanceType":"*instance-type-api-name*" Where *instance-type-api-name* is the name of the instance type (`t2.micro` | `t2.small` | `t2.medium` | `t2.large` | `m4.large` | `m4.xlarge` | `m4.2xlarge` | `m4.4xlarge` | `m4.10xlarge` | `m3.medium` | `m3.large` | `m3.xlarge` | `m3.2xlarge` | `m1.small` | `m1.medium` | `m1.large` | `m1.xlarge` | `c4.large` | `c4.xlarge` | `c4.2xlarge` | `c4.4xlarge` | `c4.8xlarge` | `c3.large` | `c3.xlarge` | `c3.2xlarge` | `c3.4xlarge` | `c3.8xlarge` | `c1.medium` | `c1.xlarge` | `cc2.8xlarge` | `r3.large` | `r3.xlarge` | `r3.2xlarge` | `r3.4xlarge` | `r3.8xlarge` | `m2.xlarge` | `m2.2xlarge` | `m2.4xlarge` | `cr1.8xlarge` | `i2.xlarge` | `i2.2xlarge` | `i2.4xlarge` | `i2.8xlarge` | `d2.xlarge` | `d2.2xlarge` | `d2.4xlarge` | `d2.8xlarge` | `hi1.4xlarge` | `hs1.8xlarge` | `t1.micro` | `g2.2xlarge` | `g2.8xlarge` | `cg1.4xlarge`).	String, Null
ec2:Owner	"ec2:Owner":"*account-id*" Where *account-id* is `amazon` | `aws-marketplace` | *aws-account-id*	String, Null
ec2:ParentSnapshot	"ec2:ParentSnapshot":"*snapshot-arn*" Where *snapshot-arn* is the snapshot ARN	ARN, Null
ec2:ParentVolume	"ec2:ParentVolume":"*volume-arn*" Where *volume-arn* is the volume ARN	ARN, Null
ec2:PlacementGroup	"ec2:PlacementGroup":"*placement-group-arn*" Where *placement-group-arn* is the placement group ARN	ARN, Null
ec2:PlacementGroupStrategy	"ec2:PlacementGroupStrategy":"*placement-group-strategy*" Where *placement-group-strategy* is `cluster`	String, Null
ec2:ProductCode	"ec2:ProductCode":"*product-code*" Where *product-code* is the product code	String, Null
ec2:Public	"ec2:Public":"*public-flag*" Where *public-flag* for an AMI is `true` | `false`	Boolean, Null
ec2:Region	"ec2:Region":"*region-name*" Where *region-name* is the name of the region (for example, `us-west-2`). To list your regions, use ec2-describe-regions.	String, Null
ec2:RequesterVpc	"ec2:RequesterVpc":"*vpc-arn*" Where *vpc-arn* is the VPC ARN for the requester's VPC	ARN, Null
ec2:ResourceTag/*tag-key*	"ec2:ResourceTag/*tag-key*":"*tag-value*" Where *tag-key* and *tag-value* are the tag-key pair	String, Null

Condition Key	Key/Value Pair	Evaluation Types
ec2:RootDevice-Type	"ec2:RootDeviceType":"*root-device-type-name*" Where *root-device-type-name* is `ebs` | `instance-store`	String, Null
ec2:Subnet	"ec2:Subnet":"*subnet-arn*" Where *subnet-arn* is the subnet ARN	ARN, Null
ec2:Tenancy	"ec2:Tenancy":"*tenancy-attribute*" Where *tenancy-attribute* is `default` | `dedicated`	String, Null
ec2:VolumeIops	"ec2:VolumeIops":"*volume-iops*" Where *volume-iops* is the input/output operations per second (IOPS); the range is 100 to 20,000	Numeric, Null
ec2:VolumeSize	"ec2:VolumeSize":"*volume-size*" Where *volume-size* is the size of the volume, in GiB	Numeric, Null
ec2:VolumeType	"ec2:VolumeType":"*volume-type-name*" Where *volume-type-name* is `gp2` for General Purpose (SSD) volumes, `standard` for Magnetic Amazon EBS volumes, or `io1` for Provisioned IOPS (SSD) volumes.	String, Null
ec2:Vpc	"ec2:Vpc":"*vpc-arn*" Where *vpc-arn* is the VPC ARN	ARN, Null

For information about which condition keys you can use with which Amazon EC2 resources, on an action-by-action basis, see Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 416). For example policy statements for Amazon EC2, see Example Policies for Working With the AWS CLI, the Amazon EC2 CLI, or an AWS SDK (p. 424).

Checking that Users Have the Required Permissions

After you've created an IAM policy, we recommend that you check whether it grants users the permissions to use the particular API actions and resources they need before you put the policy into production.

First, create an IAM user for testing purposes, and then attach the IAM policy that you created to the test user. Then, make a request as the test user.

If the action that you are testing creates or modifies a resource, you should make the request using the `DryRun` parameter (or run the CLI command with the `--auth-dry-run` option). In this case, the call completes the authorization check, but does not complete the operation. For example, you can check whether the user can terminate a particular instance without actually terminating it. If the test user has the required permissions, the request returns `DryRunOperation`; otherwise, it returns `UnauthorizedOperation`.

If the policy doesn't grant the user the permissions that you expected, or is overly permissive, you can adjust the policy as needed and retest until you get the desired results.

Important
It can take several minutes for policy changes to propagate before they take effect. Therefore, we recommend that you allow five minutes to pass before you test your policy updates.

If an authorization check fails, the request returns an encoded message with diagnostic information. You can decode the message using the `DecodeAuthorizationMessage` action. For more information, see DecodeAuthorizationMessage in the *AWS Security Token Service API Reference*, and decode-authorization-message in the *AWS Command Line Interface Reference*.

For additional information about resource-level permissions in Amazon EC2, see the following AWS Security Blog post: Demystifying EC2 Resource-Level Permissions.

Supported Resource-Level Permissions for Amazon EC2 API Actions

Resource-level permissions refers to the ability to specify which resources users are allowed to perform actions on. Amazon EC2 has partial support for resource-level permissions. This means that for certain Amazon EC2 actions, you can control when users are allowed to use those actions based on conditions that have to be fulfilled, or specific resources that users are allowed to use. For example, you can grant users permission to launch instances, but only of a specific type, and only using a specific AMI.

The following table describes the Amazon EC2 API actions that currently support resource-level permissions, as well as the supported resources (and their ARNs) and condition keys for each action.

Important
If an Amazon EC2 API action is not listed in this table, then it does not support resource-level permissions. If an Amazon EC2 API action does not support resource-level permissions, you can grant users permission to use the action, but you have to specify a * for the resource element of your policy statement. For an example of how to do this, see 1: Allow users to list the Amazon EC2 resources that belong to the AWS account (p. 425). We'll add support for additional actions, ARNs, and condition keys later. For a list of Amazon EC2 API actions that currently do not support resource-level permissions, see Unsupported Resource-Level Permissions in the *Amazon EC2 API Reference*.

API Action	Resources	Condition Keys
AcceptVpcPeeringConnection	VPC peering connection arn:aws:ec2:*region*:*account*:vpc-peering-connection/*vpc-peering-connection-id*	ec2:AccepterVpc ec2:Region ec2:ResourceTag/*tag-key* ec2:RequesterVpc
	VPC arn:aws:ec2:*region*:*account*:vpc/*vpc-id* Where *vpc-id* is a VPC owned by the accepter.	ec2:ResourceTag/*tag-key* ec2:Region ec2:Tenancy

API Action	Resources	Condition Keys
AttachClassicLinkVpc	Instance arn:aws:ec2:*region*:*account*:instance/* arn:aws:ec2:*region*:*account*:instance/*instance-id*	ec2:AvailabilityZone ec2:InstanceType ec2:PlacementGroup ec2:ProductCode ec2:Region ec2:ResourceTag/*tag-key* ec2:RootDeviceType ec2:Subnet ec2:Tenancy ec2:Vpc
	Security group arn:aws:ec2:*region*:*account*:security-group/* arn:aws:ec2:*region*:*account*:security-group/*security-group-id* Where the security group is the VPC's security group.	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
	VPC arn:aws:ec2:*region*:*account*:vpc/* arn:aws:ec2:*region*:*account*:vpc/*vpc-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Tenancy

API Action	Resources	Condition Keys
AttachVolume	Instance arn:aws:ec2:*region*:*account*:instance/*instance-id*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/*tag-key* ec2:RootDeviceType ec2:Tenancy
	Volume arn:aws:ec2:*region*:*account*:volume/*volume-id*	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:ResourceTag/*tag-key* ec2:VolumeIops ec2:VolumeSize ec2:VolumeType
AuthorizeSecurityGroupEgress	Security group arn:aws:ec2:*region*:*account*:security-group/*security-group-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
AuthorizeSecurityGroupIngress	Security group arn:aws:ec2:*region*:*account*:security-group/*security-group-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
CreateVpcPeeringConnection	VPC arn:aws:ec2:*region*:*account*:vpc/*vpc-id* Where *vpc-id* is a requester VPC.	ec2:ResourceTag/*tag-key* ec2:Region ec2:Tenancy
	VPC peering connection arn:aws:ec2:*region*:*account*:vpc-peering-connection/*	ec2:AccepterVpc ec2:Region ec2:RequesterVpc
DeleteCustomerGateway	Customer gateway arn:aws:ec2:*region*:*account*:customer-gateway/*cgw-id*	ec2:Region ec2:ResourceTag/*tag-key*

API Action	Resources	Condition Keys
DeleteDhcpOptions	DHCP options set arn:aws:ec2:*region*:*account*:dhcp-options/*dhcp-options-id*	ec2:Region ec2:ResourceTag/*tag-key*
DeleteInternetGateway	Internet gateway arn:aws:ec2:*region*:*account*:internet-gateway/*igw-id*	ec2:Region ec2:ResourceTag/*tag-key*
DeleteNetworkAcl	Network ACL arn:aws:ec2:*region*:*account*:network-acl/*nacl-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
DeleteNetworkAclEntry	Network ACL arn:aws:ec2:*region*:*account*:network-acl/*nacl-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
DeleteRoute	Route table arn:aws:ec2:*region*:*account*:route-table/*route-table-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
DeleteRouteTable	Route table arn:aws:ec2:*region*:*account*:route-table/*route-table-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
DeleteSecurityGroup	Security group arn:aws:ec2:*region*:*account*:security-group/*security-group-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
DeleteVolume	Volume arn:aws:ec2:*region*:*account*:volume/*volume-id*	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:ResourceTag/*tag-key* ec2:VolumeIops ec2:VolumeSize ec2:VolumeType
DeleteVpcPeeringConnection	VPC peering connection arn:aws:ec2:*region*:*account*:vpc-peering-connection/*vpc-peering-connection-id*	ec2:AccepterVpc ec2:Region ec2:ResourceTag/*tag-key* ec2:RequesterVpc

API Action	Resources	Condition Keys
DetachClassicLinkVpc	Instance arn:aws:ec2:*region*:*account*:instance/* arn:aws:ec2:*region*:*account*:instance/*instance-id*	ec2:AvailabilityZone ec2:InstanceType ec2:PlacementGroup ec2:ProductCode ec2:Region ec2:ResourceTag/*tag-key* ec2:RootDeviceType ec2:Subnet ec2:Tenancy ec2:Vpc
	VPC arn:aws:ec2:*region*:*account*:vpc/* arn:aws:ec2:*region*:*account*:vpc/*vpc-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Tenancy
DetachVolume	Instance arn:aws:ec2:*region*:*account*:instance/*instance-id*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/*tag-key* ec2:RootDeviceType ec2:Tenancy
	Volume arn:aws:ec2:*region*:*account*:volume/*volume-id*	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:ResourceTag/*tag-key* ec2:VolumeIops ec2:VolumeSize ec2:VolumeType
DisableVpcClassicLink	VPC arn:aws:ec2:*region*:*account*:vpc/* arn:aws:ec2:*region*:*account*:vpc/*vpc-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Tenancy

API Action	Resources	Condition Keys
EnableVpcClassicLink	VPC arn:aws:ec2:*region*:*account*:vpc/* arn:aws:ec2:*region*:*account*:vpc/*vpc-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Tenancy
RebootInstances	Instance arn:aws:ec2:*region*:*account*:instance/*instance-id*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/*tag-key* ec2:RootDeviceType ec2:Tenancy
RejectVpcPeeringConnection	VPC peering connection arn:aws:ec2:*region*:*account*:vpc-peering-connection/*vpc-peering-connection-id*	ec2:AccepterVpc ec2:Region ec2:ResourceTag/*tag-key* ec2:RequesterVpc
RevokeSecurityGroupEgress	Security group arn:aws:ec2:*region*:*account*:security-group/*security-group-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
RevokeSecurityGroupIngress	Security group arn:aws:ec2:*region*:*account*:security-group/*security-group-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc

API Action	Resources	Condition Keys
RunInstances	Image arn:aws:ec2:*region*::image/*image-id*	ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/*tag-key*
	Instance arn:aws:ec2:*region*:*account*:instance/*instance-id*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	Key pair arn:aws:ec2:*region*:*account*:key-pair/*key-pair-name*	ec2:Region
	Network interface arn:aws:ec2:*region*:*account*:network-interface/* arn:aws:ec2:*region*:*account*:network-interface/*eni-id*	ec2:AvailabilityZone ec2:Region ec2:Subnet ec2:ResourceTag/*tag-key* ec2:Vpc
	Placement group arn:aws:ec2:*region*:*account*:placement-group/*placement-group-name*	ec2:Region ec2:PlacementGroupStrategy
	Security group arn:aws:ec2:*region*:*account*:security-group/*security-group-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
	Snapshot arn:aws:ec2:*region*::snapshot/*snapshot-id*	

API Action	Resources	Condition Keys
		ec2:Owner ec2:ParentVolume ec2:Region ec2:SnapshotTime ec2:ResourceTag/*tag-key* ec2:VolumeSize
	Subnet arn:aws:ec2:*region*:*account*:subnet/*subnet-id*	ec2:AvailabilityZone ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
	Volume arn:aws:ec2:*region*:*account*:volume/*volume-id*	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:VolumeIops ec2:VolumeSize ec2:VolumeType
StartInstances	Instance arn:aws:ec2:*region*:*account*:instance/*instance-id*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/*tag-key* ec2:RootDeviceType ec2:Tenancy

API Action	Resources	Condition Keys
StopInstances	Instance arn:aws:ec2:*region*:*account*:instance/*instance-id*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/*tag-key* ec2:RootDeviceType ec2:Tenancy
TerminateInstances	Instance arn:aws:ec2:*region*:*account*:instance/*instance-id*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/*tag-key* ec2:RootDeviceType ec2:Tenancy

Example Policies for Working With the AWS CLI, the Amazon EC2 CLI, or an AWS SDK

The following examples show policy statements that you could use to control the permissions that IAM users have to Amazon EC2. These policies are designed for requests that are made with the AWS CLI, the Amazon EC2 CLI, or an AWS SDK. For example policies for working in the Amazon EC2 console, see Example Policies for Working in the Amazon EC2 Console (p. 434). For examples of IAM policies specific to Amazon VPC, see Controlling Access to Amazon VPC Resources

- 1: Allow users to list the Amazon EC2 resources that belong to the AWS account (p. 425)
- 2: Allow users to describe, launch, stop, start, and terminate all instances (p. 425)
- 3: Allow users to describe all instances, and stop, start, and terminate only particular instances (p. 425)
- 4. Allow users to manage particular volumes for particular instances (p. 426)
- 5: Allow users to launch instances with a specific configuration (p. 427)
- 6. Allow users to work with ClassicLink (p. 431)

Example 1: Allow users to list the Amazon EC2 resources that belong to the AWS account

The following policy grants users permission to use all Amazon EC2 API actions whose names begin with `Describe`. The `Resource` element uses a wildcard to indicate that users can specify all resources with these API actions. The * wildcard is also necessary in cases where the API action does not support resource-level permissions. For more information about which ARNs you can use with which Amazon EC2 API actions, see Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 416).

Users don't have permission to perform any actions on the resources (unless another statement grants them permission to do so) because they're denied permission to use API actions by default.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    }
   ]
}
```

Example 2: Allow users to describe, launch, stop, start, and terminate all instances

The following policy grants users permission to use the API actions specified in the `Action` element. The `Resource` element uses a * wildcard to indicate that users can specify all resources with these API actions. The * wildcard is also necessary in cases where the API action does not support resource-level permissions. For more information about which ARNs you can use with which Amazon EC2 API actions, see Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 416).

The users don't have permission to use any other API actions (unless another statement grants them permission to do so) because users are denied permission to use API actions by default.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances", "ec2:DescribeImages",
        "ec2:DescribeKeyPairs", "ec2:DescribeSecurityGroups",
        "ec2:DescribeAvailabilityZones",
        "ec2:RunInstances", "ec2:TerminateInstances",
        "ec2:StopInstances", "ec2:StartInstances"
      ],
      "Resource": "*"
    }
   ]
}
```

Example 3: Allow users to describe all instances, and stop, start, and terminate only particular instances

The following policy allows users to describe all instances, to start and stop only instances i-123abc12 and i-4c3b2a1, and to terminate only instances in the US East (N. Virginia) region (`us-east-1`) with the resource tag "`purpose=test`".

The first statement uses a * wildcard for the `Resource` element to indicate that users can specify all resources with the action; in this case, they can list all instances. The * wildcard is also necessary in

cases where the API action does not support resource-level permissions (in this case, `ec2:DescribeInstances`). For more information about which ARNs you can use with which Amazon EC2 API actions, see Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 416).

The second statement uses resource-level permissions for the `StopInstances` and `StartInstances` actions. The specific instances are indicated by their ARNs in the `Resource` element.

The third statement allows users to terminate all instances in the US East (N. Virginia) region (`us-east-1`) that belong to the specified AWS account, but only where the instance has the tag `"purpose=test"`. The `Condition` element qualifies when the policy statement is in effect.

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
   "Effect": "Allow",
      "Action": "ec2:DescribeInstances",
      "Resource": "*"
   },
   {
      "Effect": "Allow",
      "Action": [
        "ec2:StopInstances",
        "ec2:StartInstances"
      ],
      "Resource": [
      "arn:aws:ec2:us-east-1:123456789012:instance/i-123abc12",
      "arn:aws:ec2:us-east-1:123456789012:instance/i-4c3b2a1"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "ec2:TerminateInstances",
      "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
      "Condition": {
         "StringEquals": {
            "ec2:ResourceTag/purpose": "test"
         }
      }
   }

   ]
}
```

Example 4. Allow users to manage particular volumes for particular instances

When an API action requires a caller to specify multiple resources, you must create a policy statement that allows users to access all required resources. If you need to use a `Condition` element with one or more of these resources, you must create multiple statements as shown in this example.

The following policy allows users to attach volumes with the tag "`volume_user=`*iam-user-name*" to instances with the tag "`department=dev`", and to detach those volumes from those instances. If you attach this policy to an IAM group, the `aws:username` policy variable gives each IAM user in the group permission to attach or detach volumes from the instances with a tag named `volume_user` that has his or her IAM user name as a value.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/department": "dev"
        }
      }
   },
   {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/volume_user": "${aws:username}"
        }
      }
   }
  ]
}
```

Example 5: Allow users to launch instances with a specific configuration

The RunInstances API action launches one or more instances. `RunInstances` requires an AMI and creates an instance; and users can specify a key pair and security group in the request. Launching into EC2-VPC requires a subnet, and creates a network interface. Launching from an Amazon EBS-backed AMI creates a volume. Therefore, the user must have permission to use these Amazon EC2 resources. The caller can also configure the instance using optional parameters to `RunInstances`, such as the instance type and a subnet. You can create a policy statement that requires users to specify an optional parameter, or restricts users to particular values for a parameter. The examples in this section demonstrate some of the many possible ways that you can control the configuration of an instance that a user can launch.

Note that by default, users don't have permission to describe, start, stop, or terminate the resulting instances. One way to grant the users permission to manage the resulting instances is to create a specific tag for each instance, and then create a statement that enables them to manage instances with that tag. For more information, see Example 3: Allow users to stop and start only particular instances (p. 425).

a. AMI

The following policy allows users to launch instances using only the AMIs that have the specified tag, "`department=dev`", associated with them. The users can't launch instances using other AMIs because the `Condition` element of the first statement requires that users specify an AMI that has this tag. The users also can't launch into a subnet, as the policy does not grant permissions for the subnet and network interface resources. They can, however, launch into EC2-Classic. The second statement uses a wildcard to enable users to create instance resources, and requires users to specify the key pair `project_keypair` and the security group `sg-1a2b3c4d`. Users are still able to launch instances without a key pair.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:region::image/ami-*"
      ],
      "Condition": {
         "StringEquals": {
            "ec2:ResourceTag/department": "dev"
         }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
          "arn:aws:ec2:region:account:instance/*",
          "arn:aws:ec2:region:account:volume/*",
          "arn:aws:ec2:region:account:key-pair/project_keypair",
          "arn:aws:ec2:region:account:security-group/sg-1a2b3c4d"
         ]
      }
   ]
}
```

Alternatively, the following policy allows users to launch instances using only the specified AMIs, `ami-9e1670f7` and `ami-45cf5c3c`. The users can't launch an instance using other AMIs (unless another statement grants the users permission to do so), and the users can't launch an instance into a subnet.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region::image/ami-9e1670f7",
        "arn:aws:ec2:region::image/ami-45cf5c3c",
        "arn:aws:ec2:region:account:instance/*",
        "arn:aws:ec2:region:account:volume/*",
        "arn:aws:ec2:region:account:key-pair/*",
        "arn:aws:ec2:region:account:security-group/*"
      ]
    }
   ]
}
```

Alternatively, the following policy allows users to launch instances from all AMIs owned by Amazon. The `Condition` element of the first statement tests whether `ec2:Owner` is `amazon`. The users can't launch an instance using other AMIs (unless another statement grants the users permission to do so). The users are able to launch an instance into a subnet.

```
{
   "Version": "2012-10-17",
   "Statement": [{
```

```
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:region::image/ami-*"
      ],
      "Condition": {
         "StringEquals": {
            "ec2:Owner": "amazon"
            }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:region:account:instance/*",
         "arn:aws:ec2:region:account:subnet/*",
         "arn:aws:ec2:region:account:volume/*",
         "arn:aws:ec2:region:account:network-interface/*",
         "arn:aws:ec2:region:account:key-pair/*",
         "arn:aws:ec2:region:account:security-group/*"
         ]
      }
   ]
}
```

b. Instance type

The following policy allows users to launch instances using only the `t2.micro` or `t2.small` instance type, which you might do to control costs. The users can't launch larger instances because the `Condition` element of the first statement tests whether `ec2:InstanceType` is either `t2.micro` or `t2.small`.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:region:account:instance/*"
      ],
      "Condition": {
         "StringEquals": {
            "ec2:InstanceType": ["t2.micro", "t2.small"]
         }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:region::image/ami-*",
         "arn:aws:ec2:region:account:subnet/*",
         "arn:aws:ec2:region:account:network-interface/*",
         "arn:aws:ec2:region:account:volume/*",
         "arn:aws:ec2:region:account:key-pair/*",
         "arn:aws:ec2:region:account:security-group/*"
         ]
```

```
      }
   ]
}
```

Alternatively, you can create a policy that denies users permission to launch any instances except `t2.micro` and `t2.small` instance types.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:region:account:instance/*"
      ],
      "Condition": {
         "StringNotEquals": {
            "ec2:InstanceType": ["t2.micro", "t2.small"]
            }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:region::image/ami-*",
         "arn:aws:ec2:region:account:network-interface/*",
         "arn:aws:ec2:region:account:instance/*",
         "arn:aws:ec2:region:account:subnet/*",
         "arn:aws:ec2:region:account:volume/*",
         "arn:aws:ec2:region:account:key-pair/*",
         "arn:aws:ec2:region:account:security-group/*"
         ]
      }
   ]
}
```

c. Subnet

The following policy allows users to launch instances using only the specified subnet, `subnet-12345678`. The group can't launch instances into any another subnet (unless another statement grants the users permission to do so). Users are still able to launch instances into EC2-Classic.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region:account:subnet/subnet-12345678",
        "arn:aws:ec2:region:account:network-interface/*",
        "arn:aws:ec2:region:account:instance/*",
        "arn:aws:ec2:region:account:volume/*",
        "arn:aws:ec2:region::image/ami-*",
        "arn:aws:ec2:region:account:key-pair/*",
        "arn:aws:ec2:region:account:security-group/*"
```

```
      ]
    }
   ]
}
```

Alternatively, you could create a policy that denies users permission to launch an instance into any other subnet. The statement does this by denying permission to create a network interface, except where subnet `subnet-12345678` is specified. This denial overrides any other policies that are created to allow launching instances into other subnets. Users are still able to launch instances into EC2-Classic.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:region:account:network-interface/*"
      ],
      "Condition": {
         "ArnNotEquals": {
            "ec2:Subnet": "arn:aws:ec2:region:account:subnet/subnet-12345678"
            }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:region::image/ami-*",
         "arn:aws:ec2:region:account:network-interface/*",
         "arn:aws:ec2:region:account:instance/*",
         "arn:aws:ec2:region:account:subnet/*",
         "arn:aws:ec2:region:account:volume/*",
         "arn:aws:ec2:region:account:key-pair/*",
         "arn:aws:ec2:region:account:security-group/*"
         ]
      }
   ]
}
```

Example 6. Allow users to work with ClassicLink

You can enable a VPC for ClassicLink and then link an EC2-Classic instance to the VPC. You can also view your ClassicLink-enabled VPCs, and all of your EC2-Classic instances that are linked to a VPC. You can create policies with resource-level permission for the `ec2:EnableVpcClassicLink`, `ec2:DisableVpcClassicLink`, `ec2:AttachClassicLinkVpc`, and `ec2:DetachClassicLinkVpc` actions to control how users are able to use those actions. Resource-level permissions are not supported for `ec2:Describe*` actions.

a. Full permission to work with ClassicLink

The following policy grants users permission to view ClassicLink-enabled VPCs and linked EC2-Classic instances, to enable and disable a VPC for ClassicLink, and to link and unlink instances from a ClassicLink-enabled VPC.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeClassicLinkInstances", "ec2:DescribeVpcClassicLink",
        "ec2:EnableVpcClassicLink", "ec2:DisableVpcClassicLink",
        "ec2:AttachClassicLinkVpc", "ec2:DetachClassicLinkVpc"
      ],
      "Resource": "*"
    }
   ]
}
```

b. Enable and disable a VPC for ClassicLink

The following policy allows user to enable and disable VPCs for ClassicLink that have the specific tag '`purpose=classiclink`'. Users cannot enable or disable any other VPCs for ClassicLink.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*VpcClassicLink",
      "Resource": "arn:aws:ec2:region:account:vpc/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/purpose":"classiclink"
        }
      }
    }
  ]
}
```

c. Link instances

The following policy grants users permission to link instances to a VPC only if the instance is an `m3.large` instance type. The second statement allows users to use the VPC and security group resources, which are required to link an instance to a VPC.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:AttachClassicLinkVpc",
      "Resource": "arn:aws:ec2:region:account:instance/*",
      "Condition": {
        "StringEquals": {
          "ec2:InstanceType":"m3.large"
        }
      }
    },
    {
      "Effect": "Allow",
```

```
      "Action": "ec2:AttachClassicLinkVpc",
      "Resource": [
        "arn:aws:ec2:region:account:vpc/*",
        "arn:aws:ec2:region:account:security-group/*"
      ]
    }
  ]
}
```

The following policy grants users permission to link instances to a specific VPC (`vpc-1a2b3c4d`) only, and to associate only specific security groups from the VPC to the instance (`sg-1122aabb` and `sg-aabb2233`). Users cannot link an instance to any other VPC, and they cannot specify any other of the VPC's security groups to associate with the instance in the request.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:AttachClassicLinkVpc",
      "Resource": [
         "arn:aws:ec2:region:account:vpc/vpc-1a2b3c4d",
         "arn:aws:ec2:region:account:instance/*",
         "arn:aws:ec2:region:account:security-group/sg-1122aabb",
         "arn:aws:ec2:region:account:security-group/sg-aabb2233"
       ]
    }
  ]
}
```

d. Unlink instances

The following grants users permission to unlink any linked EC2-Classic instance from a VPC, but only if the instance has the tag "`unlink=true`". The second statement grants users permission to use the VPC resource, which is required to unlink an instance from a VPC.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "ec2:DetachClassicLinkVpc",
      "Resource": [
         "arn:aws:ec2:region:account:instance/*"
      ],
      "Condition": {
         "StringEquals": {
            "ec2:ResourceTag/unlink":"true"
            }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:DetachClassicLinkVpc",
      "Resource": [
         "arn:aws:ec2:region:account:vpc/*"
         ]
```

```
      }
   ]
}
```

Example Policies for Working in the Amazon EC2 Console

You can use IAM policies to grant users permissions to view and work with specific resources in the Amazon EC2 console. You can use the example policies in the previous section; however, they are designed for requests that are made with the AWS CLI, the Amazon EC2 CLI, or an AWS SDK. The console uses additional API actions for its features, so these policies may not work as expected. For example, a user that has permission to use only the `DescribeVolumes` API action will encounter errors when trying to view volumes in the console. This section demonstrates policies that enable users to work with specific parts of the console.

- 1: Read-only access (p. 435)
- 2: Using the EC2 launch wizard (p. 436)
- 3: Working with volumes (p. 439)
- 4: Working with security groups (p. 440)
- 5: Working with Elastic IP addresses (p. 442)

> **Note**
> To help you work out which API actions are required to perform tasks in the console, you can use a service such as AWS CloudTrail. For more information, see the AWS CloudTrail User Guide. If your policy does not grant permission to create or modify a specific resource, the console displays an encoded message with diagnostic information. You can decode the message using the DecodeAuthorizationMessage API action for AWS STS, or the decode-authorization-message command in the AWS CLI.

For additional information about creating policies for the Amazon EC2 console, see the following AWS Security Blog post: Granting Users Permission to Work in the Amazon EC2 Console.

Example 1: Read-only access

To allow users to view all resources in the Amazon EC2 console, you can use the same policy as the following example: 1: Allow users to list the Amazon EC2 resources that belong to the AWS account (p. 425). Users cannot perform any actions on those resources or create new resources, unless another statement grants them permission to do so.

a. View instances, AMIs, and snapshots

Alternatively, you can provide read-only access to a subset of resources. To do this, replace the * wildcard in the `ec2:Describe` API action with specific `ec2:Describe` actions for each resource. The following policy allows users to view all instances, AMIs, and snapshots in the Amazon EC2 console. The `ec2:DescribeTags` action allows users to view public AMIs. The console requires the tagging information to display public AMIs; however, you can remove this action if you want users to view only private AMIs.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeInstances", "ec2:DescribeImages",
         "ec2:DescribeTags", "ec2:DescribeSnapshots"
      ],
      "Resource": "*"
   }
   ]
}
```

Note
Currently, the Amazon EC2 `ec2:Describe*` API actions do not support resource-level permissions, so you cannot control which individual resources users can view in the console. Therefore, the * wildcard is necessary in the `Resource` element of the above statement. For more information about which ARNs you can use with which Amazon EC2 API actions, see Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 416).

b. View instances and CloudWatch metrics

The following policy allows users to view instances in the Amazon EC2 console, as well as view CloudWatch alarms and metrics in the **Monitoring** tab of the **Instances** page. The Amazon EC2 console uses the Amazon CloudWatch APIs to display the alarms and metrics, so you must grant users permission to use the `cloudwatch:DescribeAlarms` and `cloudwatch:GetMetricStatistics` actions.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeInstances",
         "cloudwatch:DescribeAlarms",
         "cloudwatch:GetMetricStatistics"
      ],
      "Resource": "*"
   }
   ]
}
```

Example 2: Using the EC2 launch wizard

The Amazon EC2 launch wizard is a series of screens with options to configure and launch an instance. Your policy must include permission to use the API actions that allow users to work with the wizard's options. If your policy does not include permission to use those actions, some items in the wizard cannot load properly, and users cannot complete a launch.

a. Basic launch wizard access

To complete a launch successfully, users must be given permission to use the `ec2:RunInstances` API action, and at least the following API actions:

- `ec2:DescribeImages`: To view and select an AMI.
- `ec2:DescribeVPCs`: To view the available network options, which are EC2-Classic and a list of VPCs. This is required even if you are not launching into a VPC.
- `ec2:DescribeSubnets`: If launching into a VPC, to view all available subnets for the chosen VPC.
- `ec2:DescribeSecurityGroups`: To view the security groups page in the wizard. Users can select an existing security group.
- `ec2:DescribeKeyPairs` or `ec2:CreateKeyPair`: To select an existing key pair, or create a new one.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
  "ec2:DescribeInstances", "ec2:DescribeImages",
        "ec2:DescribeKeyPairs","ec2:DescribeVpcs", "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*"
    }
   ]
}
```

You can add API actions to your policy to provide more options for users, for example:

- `ec2:DescribeAvailabilityZones`: If launching into EC2-Classic, to view and select a specific Availability Zone.
- `ec2:DescribeNetworkInterfaces`: If launching into a VPC, to view and select existing network interfaces for the selected subnet.
- `ec2:CreateSecurityGroup`: To create a new security group; for example, to create the wizard's suggested `launch-wizard-x` security group. However, this action alone only creates the security group; it does not add or modify any rules. To add inbound rules, users must be granted permission to use the `ec2:AuthorizeSecurityGroupIngress` API action. To add outbound rules to VPC security groups, users must be granted permission to use the `ec2:AuthorizeSecurityGroupEgress` API action. To modify or delete existing rules, users must be granted permission to use the relevant `ec2:RevokeSecurityGroup*` API action.
- `ec2:CreateTags`: To add a tag to the instance. By default, the launch wizard attempts to add a tag with a key of `Name` to an instance. Users that do not have permission to use this action will encounter

a warning that this tag could not be applied to an instance; however, this does not affect the success of the launch, so you should only grant users permission to use this action if it's absolutely necessary.

Important
Be careful about granting users permission to use the `ec2:CreateTags` action. This limits your ability to use the `ec2:ResourceTag` condition key to restrict the use of other resources; users can change a resource's tag in order to bypass those restrictions.

Currently, the Amazon EC2 `Describe*` API actions do not support resource-level permissions, so you cannot restrict which individual resources users can view in the launch wizard. However, you can apply resource-level permissions on the `ec2:RunInstances` API action to restrict which resources users can use to launch an instance. The launch fails if users select options that they are not authorized to use.

b. Restrict access to specific instance type, subnet, and region

The following policy allows users to launch `m1.small` instances using AMIs owned by Amazon, and only into a specific subnet (`subnet-1a2b3c4d`). Users can only launch in the sa-east-1 region. If users select a different region, or select a different instance type, AMI, or subnet in the launch wizard, the launch fails.

The first statement grants users permission to view the options in the launch wizard, as demonstrated in the example above. The second statement grants users permission to use the network interface, volume, key pair, security group, and subnet resources for the `ec2:RunInstances` action, which are required to launch an instance into a VPC. For more information about using the `ec2:RunInstances` action, see 5: Allow users to launch instances with a specific configuration (p. 427). The third and fourth statements grant users permission to use the instance and AMI resources respectively, but only if the instance is an `m1.small` instance, and only if the AMI is owned by Amazon.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeInstances", "ec2:DescribeImages",
         "ec2:DescribeKeyPairs","ec2:DescribeVpcs", "ec2:DescribeSubnets",
"ec2:DescribeSecurityGroups"
   ],
   "Resource": "*"
   },
   {
      "Effect": "Allow",
      "Action":"ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:sa-east-1:111122223333:network-interface/*",
         "arn:aws:ec2:sa-east-1:111122223333:volume/*",
         "arn:aws:ec2:sa-east-1:111122223333:key-pair/*",
         "arn:aws:ec2:sa-east-1:111122223333:security-group/*",
         "arn:aws:ec2:sa-east-1:111122223333:subnet/subnet-1a2b3c4d"
      ]
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:sa-east-1:111122223333:instance/*"
      ],
      "Condition": {
         "StringEquals": {
            "ec2:InstanceType": "m1.small"
         }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
            "arn:aws:ec2:sa-east-1::image/ami-*"
      ],
      "Condition": {
         "StringEquals": {
            "ec2:Owner": "amazon"
         }
      }
   }
   ]
}
```

Example 3: Working with volumes

The following policy grants users permission to view and create volumes, and attach and detach volumes to specific instances.

Users can attach any volume to instances that have the tag "`purpose=test`", and also detach volumes from those instances. To attach a volume using the Amazon EC2 console, it is helpful for users to have permission to use the `ec2:DescribeInstances` action, as this allows them to select an instance from a pre-populated list in the **Attach Volume** dialog box. However, this also allows users to view all instances on the **Instances** page in the console, so you can omit this action.

In the first statement, the `ec2:DescribeVolumeStatus` and `ec2:DescribeAvailabilityZones` actions are necessary to ensure that volumes display correctly in the console.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVolumes", "ec2:DescribeVolumeStatus",
  "ec2:DescribeAvailabilityZones", "ec2:CreateVolume",
  "ec2:DescribeInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Resource": "arn:aws:ec2:region:111122223333:instance/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/purpose": "test"
        }
     }
   },
   {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Resource": "arn:aws:ec2:region:111122223333:volume/*"
 }
   ]
}
```

Example 4: Working with security groups

a. View security groups and add and remove rules

The following policy grants users permission to view security groups in the Amazon EC2 console, and to add and remove inbound and outbound rules for existing security groups that have the tag `Department=Test`.

> **Note**
> You can't modify outbound rules for EC2-Classic security groups. For more information about security groups, see Amazon EC2 Security Groups for Windows Instances (p. 398).

In the first statement, the `ec2:DescribeTags` action allows users to view tags in the console, which makes it easier for users to identify the security groups that they are allowed to modify.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeSecurityGroups", "ec2:DescribeTags"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AuthorizeSecurityGroupIngress", "ec2:RevokeSecurityGroupIngress",

         "ec2:AuthorizeSecurityGroupEgress", "ec2:RevokeSecurityGroupEgress"
      ],
      "Resource": [
         "arn:aws:ec2:region:111122223333:security-group/*"
      ],
      "Condition": {
         "StringEquals": {
            "ec2:ResourceTag/Department": "Test"
         }
      }
   }
   ]
}
```

b. Working with the Create Security Group dialog box

You can create a policy that allows users to work with the **Create Security Group** dialog box in the Amazon EC2 console. To use this dialog box, users must be granted permission to use at the least the following API actions:

- `ec2:CreateSecurityGroup`: To create a new security group.
- `ec2:DescribeVpcs`: To view a list of existing VPCs in the **VPC** list. This action is required even if you are not creating a security group for a VPC.

With these permissions, users can create a new security group successfully, but they cannot add any rules to it. To work with rules in the **Create Security Group** dialog box, you can add the following API actions to your policy:

- `ec2:AuthorizeSecurityGroupIngress`: To add inbound rules.

- `ec2:AuthorizeSecurityGroupEgress`: To add outbound rules to VPC security groups.
- `ec2:RevokeSecurityGroupIngress`: To modify or delete existing inbound rules. This is useful if you want to allow users to use the **Copy to new** feature in the console. This feature opens the **Create Security Group** dialog box and populates it with the same rules as the security group that was selected.
- `ec2:RevokeSecurityGroupEgress`: To modify or delete outbound rules for VPC security groups. This is useful to allow users to modify or delete the default outbound rule that allows all outbound traffic.
- `ec2:DeleteSecurityGroup`: To cater for scenarios where invalid rules cannot be saved. If a user creates a security group with an invalid rule, the console first creates the security group, then attempts to add the rules to it. After that fails, the security group is deleted. The user remains in the **Create Security Group** dialog box, where an error is displayed. The rules remain listed, so the user can correct the invalid rule and try to create the security group again. This API action is not required, but if a user is not granted permission to use it and attempts to create a security group with invalid rules, the security group is created without any rules, and the user must add them afterward.

Currently, the `ec2:CreateSecurityGroup` API action does not support resource-level permissions; however, you can apply resource-level permissions to the `ec2:AuthorizeSecurityGroupIngress` and `ec2:AuthorizeSecurityGroupEgress` actions to control how users can create rules.

The following policy grants users permission to use the **Create Security Group** dialog box, and to create inbound and outbound rules for security groups that are associated with a specific VPC (`vpc-1a2b3c4d`). Users can create security groups for EC2-Classic or another VPC, but they cannot add any rules to them. Similarly, users cannot add any rules to any existing security group that's not associated with VPC `vpc-1a2b3c4d`. Users are also granted permission to view all security groups in the console. This makes it easier for users to identify the security groups to which they can add inbound rules. This policy also grants users permission to delete security groups that are associated with VPC `vpc-1a2b3c4d`.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSecurityGroups", "ec2:CreateSecurityGroup", "ec2:De
scribeVpcs"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteSecurityGroup", "ec2:AuthorizeSecurityGroupIngress",
"ec2:AuthorizeSecurityGroupEgress"
      ],
      "Resource": "arn:aws:ec2:region:111122223333:security-group/*",
      "Condition":{
         "ArnEquals": {
            "ec2:Vpc": "arn:aws:ec2:region:111122223333:vpc/vpc-1a2b3c4d"
         }
      }
    }
   ]
}
```

Example 5: Working with Elastic IP addresses

The following policy grants users permission to view Elastic IP addresses in the Amazon EC2 console. The console uses the `ec2:DescribeInstances` action to display information about instances with which the Elastic IP addresses are associated. If users are not granted permission to use this action, the Elastic IP addresses page cannot load properly.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeAddresses", "ec2:DescribeInstances"
      ],
      "Resource": "*"
   }
   ]
}
```

To allow users to work with Elastic IP addresses, you can add the following actions to your policy

- `ec2:AllocateAddress`: To allocate an address for use in VPC or EC2-Classic.
- `ec2:ReleaseAddress`: To release an Elastic IP address.
- `ec2:DescribeNetworkInterfaces`: To work with the **Associate Address** dialog box. The dialog box displays the available network interfaces to which you can associate an Elastic IP address, and will not open if users are not granted permission to use this action. However, this only applies to EC2-VPC; this action is not required for associating an Elastic IP address to an instance in EC2-Classic.
- `ec2:AssociateAddress`: To associate an Elastic IP address with an instance or a network interface.
- `ec2:DisassociateAddress`: To disassociate an Elastic IP address from an instance or a network interface,

IAM Roles for Amazon EC2

Applications must sign their API requests with AWS credentials. Therefore, if you are an application developer, you need a strategy for managing credentials for your applications that run on EC2 instances. For example, you can securely distribute your AWS credentials to the instances, enabling the applications on those instances to use your credentials to sign requests, while protecting them from other users. However, it's challenging to securely distribute credentials to each instance, especially those that AWS creates on your behalf, such as Spot instances or instances in Auto Scaling groups. You must also be able to update the credentials on each instance when you rotate your AWS credentials.

We designed IAM roles so that your applications can securely make API requests from your instances, without requiring you to manage the security credentials that the applications use. Instead of creating and distributing your AWS credentials, you can delegate permission to make API requests using IAM roles as follows:

1. Create an IAM role.
2. Define which accounts or AWS services can assume the role.
3. Define which API actions and resources the application can use after assuming the role.
4. Specify the role when you launch your instances.
5. Have the application retrieve a set of temporary credentials and use them.

For example, you can use IAM roles to grant permissions to applications running on your instances that needs to use a bucket in Amazon S3.

> **Note**
> Amazon EC2 uses an *instance profile* as a container for an IAM role. When you create an IAM role using the console, the console creates an instance profile automatically and gives it the same name as the role it corresponds to. If you use the AWS CLI, API, or an AWS SDK to create a role, you create the role and instance profile as separate actions, and you might give them different names. To launch an instance with an IAM role, you specify the name of its instance profile. When you launch an instance using the Amazon EC2 console, you can select a role to associate with the instance; however, the list that's displayed is actually a list of instance profile names. For more information, see Instance Profiles in the *IAM User Guide*.

You can specify permissions for IAM roles by creating a policy in JSON format. These are similar to the policies that you create for IAM users. If you make a change to a role, the change is propagated to all instances, simplifying credential management.

> **Note**
> You can't assign a role to an existing instance; you can only specify a role when you launch a new instance.

For more information about creating and using IAM roles, see Roles in the *IAM User Guide*.

Topics

- Retrieving Security Credentials from Instance Metadata (p. 443)
- Granting an IAM User Permission to Launch an Instance with an IAM Role (p. 444)
- Creating an IAM Role Using the Console (p. 444)
- Launching an Instance with an IAM Role Using the Console (p. 445)
- Creating an IAM Role Using the AWS CLI (p. 445)
- Launching an Instance with an IAM Role Using the AWS CLI (p. 447)
- Launching an Instance with an IAM Role Using an AWS SDK (p. 448)

Retrieving Security Credentials from Instance Metadata

An application on the instance retrieves the security credentials provided by the role from the instance metadata item `iam/security-credentials/`*role-name*. The application is granted the permissions for the actions and resources that you've defined for the role through the security credentials associated with the role. These security credentials are temporary and we rotate them automatically. We make new credentials available at least five minutes prior to the expiration of the old credentials.

> **Warning**
> If you use services that use instance metadata with IAM roles, ensure that you don't expose your credentials when the services make HTTP calls on your behalf. The types of services that could expose your credentials include HTTP proxies, HTML/CSS validator services, and XML processors that support XML inclusion.

The following command retrieves the security credentials for an IAM role named `s3access`.

```
C:\> curl http://169.254.169.254/latest/meta-data/iam/security-credentials/s3access
```

The following is example output.

```
{
  "Code" : "Success",
```

```
  "LastUpdated" : "2012-04-26T16:39:16Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "AKIAIOSFODNN7EXAMPLE",
  "SecretAccessKey" : "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
  "Token" : "token",
  "Expiration" : "2012-04-27T22:39:16Z"
}
```

For more information about instance metadata, see Instance Metadata and User Data (p. 160). For more information about temporary credentials, see the *Using Temporary Security Credentials*.

Granting an IAM User Permission to Launch an Instance with an IAM Role

To enable an IAM user to launch an instance with an IAM role, you must grant the user permission to pass the role to the instance.

For example, the following IAM policy grants users permission to launch an instance with the IAM role named `s3access`.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "iam:PassRole",
        "Resource": "arn:aws:iam::123456789012:role/s3access"
    }]
}
```

Alternatively, you could grant IAM users access to all your roles by specifying the resource as "*" in this policy. However, consider whether users who launch instances with your roles (ones that exist or that you'll create later on) might be granted permissions that they don't need or shouldn't have.

For more information, see Permissions Required for Using Roles with Amazon EC2 in the *IAM User Guide*.

Creating an IAM Role Using the Console

You must create an IAM role before you can launch an instance with that role.

To create an IAM role using the IAM console

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, click **Roles**, and then click **Create New Role**.
3. On the **Set Role Name** page, enter a name for the role and click **Next Step**.
4. On the **Select Role Type** page, click **Select** next to **Amazon EC2**.
5. On the **Attach Policy** page, select an AWS managed policy. For example, for Amazon EC2, one of the following AWS managed policies might meet your needs:

   - PowerUserAccess
   - ReadOnlyAccess
   - AmazonEC2FullAccess
   - AmazonEC2ReadOnlyAccess

6. Review the role information, edit the role as needed, and then click **Create Role**.

Launching an Instance with an IAM Role Using the Console

After you've created an IAM role, you can launch an instance, and associate that role with the instance during launch.

> **Important**
> After you create an IAM role, it may take several seconds for the permissions to propagate. If your first attempt to launch an instance with a role fails, wait a few seconds before trying again. For more information, see Troubleshooting Working with Roles in the *IAM User Guide*.

To launch an instance with an IAM role

1. Open the Amazon EC2 console.
2. On the dashboard, click **Launch Instance**.
3. Select an AMI, then select an instance type and click **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, select the IAM role you created from the **IAM role** list.

   > **Note**
   > The **IAM role** list displays the name of the instance profile that you created when you created your IAM role. If you created your IAM role using the console, the instance profile was created for you and given the same name as the role. If you created your IAM role using the AWS CLI, API, or an AWS SDK, you may have named your instance profile differently.

5. Configure any other details, then follow the instructions through the rest of the wizard, or click **Review and Launch** to accept default settings and go directly to the **Review Instance Launch** page.
6. Review your settings, then click **Launch** to choose a key pair and launch your instance.
7. If you are using the Amazon EC2 API actions in your application, retrieve the AWS security credentials made available on the instance and use them to sign the requests. Note that the AWS SDK does this for you.

```
C:\> curl http://169.254.169.254/latest/meta-data/iam/security-credentials/role_name
```

Creating an IAM Role Using the AWS CLI

You must create an IAM role before you can launch an instance with that role.

To create an IAM role using the AWS CLI

- Create an IAM role with a policy that allows the role to use an Amazon S3 bucket.

  a. Create the following trust policy and save it in a text file named `ec2-role-trust-policy.json`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": "ec2.amazonaws.com"},
      "Action": "sts:AssumeRole"
    }
```

```
  ]
}
```

b. Create the `s3access` role. You'll specify the trust policy you created.

```
C:\> aws iam create-role --role-name s3access --assume-role-policy-docu
ment file://ec2-role-trust-policy.json
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": "sts:AssumeRole",
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "ec2.amazonaws.com"
                    }
                }
            ]
        },
        "RoleId": "AROAIIZKPBKS2LEXAMPLE",
        "CreateDate": "2013-12-12T23:46:37.247Z",
        "RoleName": "s3access",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/s3access"
    }
}
```

c. Create an access policy and save it in a text file named `ec2-role-access-policy.json`. For example, this policy grants administrative permissions for Amazon S3 to applications running on the instance.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:*"],
      "Resource": ["*"]
    }
  ]
}
```

d. Attach the access policy to the role.

```
C:\> aws iam put-role-policy --role-name s3access --policy-name S3-Per
missions --policy-document file://ec2-role-access-policy.json
```

e. Create an instance profile named `s3access-profile`.

```
C:\> aws iam create-instance-profile --instance-profile-name S3-Permis
sions
```

```
{
    "InstanceProfile": {
        "InstanceProfileId": "AIPAJTLBPJLEGREXAMPLE",
        "Roles": [],
        "CreateDate": "2013-12-12T23:53:34.093Z",
        "InstanceProfileName": "S3-Permissions",
        "Path": "/",
       "Arn": "arn:aws:iam::123456789012:instance-profile/S3-Permissions"

    }
}
```

f. Add the `s3access` role to the `s3access-profile` instance profile.

```
C:\> aws iam add-role-to-instance-profile --instance-profile-name S3-
Permissions --role-name s3access
```

For more information about these commands, see create-role, put-role-policy, and create-instance-profile in the *AWS Command Line Interface Reference*.

Launching an Instance with an IAM Role Using the AWS CLI

After you've created an IAM role, you can launch an instance, and associate that role with the instance during launch.

Important
After you create an IAM role, it may take several seconds for the permissions to propagate. If your first attempt to launch an instance with a role fails, wait a few seconds before trying again. For more information, see Troubleshooting Working with Roles in the *IAM User Guide*.

To launch an instance with an IAM role using the AWS CLI

1. Launch an instance using the instance profile. The following example shows how to launch an instance with the instance profile.

```
C:\> aws ec2 run-instances --image-id ami-11aa22bb --iam-instance-profile
Name="S3-Permissions" --key-name my-key-pair --security-groups my-security-
group --subnet-id subnet-1a2b3c4d
```

   For more information, see run-instances in the *AWS Command Line Interface Reference*.
2. If you are using the Amazon EC2 API actions in your application, retrieve the AWS security credentials made available on the instance and use them to sign the requests. Note that the AWS SDK does this for you.

```
C:\> curl http://169.254.169.254/latest/meta-data/iam/security-creden
tials/role_name
```

Launching an Instance with an IAM Role Using an AWS SDK

If you use an AWS SDK to write your application, you automatically get temporary security credentials from the role associated with the current instance. For more information, see the following topics in the SDK documentation:

- Using IAM Roles for EC2 Instances with the SDK for Java
- Using IAM Roles for EC2 Instances with the SDK for .NET
- Using IAM Roles for EC2 Instances with the SDK for PHP
- Using IAM Roles for EC2 Instances with the SDK for Ruby

Authorizing Inbound Traffic for Your Windows Instances

Security groups enable you to control traffic to your instance, including the kind of traffic that can reach your instance. For example, you can allow computers from only your home network to access your instance using RDP. If your instance is a web server, you can allow all IP addresses to access your instance via HTTP, so that external users can browse the content on your web server.

To enable network access to your instance, you must allow inbound traffic to your instance. To open a port for inbound traffic, add a rule to a security group that you associated with your instance when you launched it.

To connect to your instance, you must set up a rule to authorize RDP traffic from your computer's public IP address. To allow RDP traffic from additional IP address ranges, add another rule for each range you need to authorize.

If you need to enable network access to a Linux instance, see Authorizing Inbound Traffic for Your Linux Instances in the *Amazon EC2 User Guide for Linux Instances*.

Before You Start

Decide who requires access to your instance; for example, a single host or a specific network that you trust. In this case, we use your local system's public IP address. You can get the public IP address of your local computer using a service. For example, we provide the following service: http://checkip.amazonaws.com. To locate another service that provides your IP address, use the search phrase "what is my IP address". If you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

> **Caution**
> If you use `0.0.0.0/0`, you enable all IP addresses to access your instance using RDP. This is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, you'll authorize only a specific IP address or range of addresses to access your instance.

For more information about security groups, see Amazon EC2 Security Groups for Windows Instances (p. 398).

Adding a Rule for Inbound RDP Traffic to a Windows Instance

Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level. You must add rules to a security group that enable you to connect to your Windows instance from your IP address using RDP.

To add a rule to a security group for inbound RDP traffic using the console

1. In the navigation pane of the Amazon EC2 console, click **Instances**. Select your instance and look at the **Description** tab; **Security groups** lists the security groups that are associated with the instance. Click **view rules** to display a list of the rules that are in effect for the instance.
2. In the navigation pane, click **Security Groups**. Select one of the security groups associated with your instance.
3. In the details pane, on the **Inbound** tab, click **Edit**. In the dialog, click **Add Rule**, and then select **RDP** from the **Type** list.
4. In the **Source** field, specify the public IP address of your computer, in CIDR notation. For example, if your IP address is `203.0.113.25`, specify `203.0.113.25/32` to list this single IP address in CIDR notation. If your company allocates addresses from a range, specify the entire range, such as `203.0.113.0/24`.

   For information about finding your IP address, see Before You Start (p. 448).
5. Click **Save**.

To add a rule to a security group using the command line

You can use one of the following commands. Be sure to run this command on your local system, not on the instance itself. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- authorize-security-group-ingress (AWS CLI)
- ec2-authorize (Amazon EC2 CLI)
- Grant-EC2SecurityGroupIngress (AWS Tools for Windows PowerShell)

Assigning a Security Group to an Instance

You can assign a security group to an instance when you launch the instance. When you add or remove rules, those changes are automatically applied to all instances to which you've assigned the security group.

After you launch an instance in EC2-Classic, you can't change its security groups. After you launch an instance in a VPC, you can change its security groups. For more information, see Changing an Instance's Security Groups in the *Amazon VPC User Guide*.

Amazon EC2 and Amazon Virtual Private Cloud

Amazon Virtual Private Cloud (Amazon VPC) enables you to define a virtual network in your own logically isolated area within the AWS cloud, known as a *virtual private cloud (VPC)*. You can launch your AWS resources, such as instances, into your VPC. Your VPC closely resembles a traditional network that you might operate in your own data center, with the benefits of using AWS's scalable infrastructure. You can configure your VPC; you can select its IP address range, create subnets, and configure route tables, network gateways, and security settings. You can connect instances in your VPC to the Internet. You can connect your VPC to your own corporate data center, making the AWS cloud an extension of your data center. To protect the resources in each subnet, you can use multiple layers of security, including security groups and network access control lists. For more information, see the Amazon VPC User Guide.

Your account may support both the EC2-VPC and EC2-Classic platforms, on a region-by-region basis. If you created your account after 2013-12-04, it supports EC2-VPC only. To find out which platforms your account supports, see Supported Platforms (p. 455). If your accounts supports EC2-VPC only, we create a *default VPC* for you. A default VPC is a VPC that is already configured and ready for you to use. You

can launch instances into your default VPC immediately. For more information, see Your Default VPC and Subnets in the *Amazon VPC User Guide*. If your account supports EC2-Classic and EC2-VPC, you can launch instances into either platform. Regardless of which platforms your account supports, you can create your own *nondefault VPC*, and configure it as you need.

Contents

- Benefits of Using a VPC (p. 450)
- Differences Between EC2-Classic and EC2-VPC (p. 450)
- Sharing and Accessing Resources Between EC2-Classic and EC2-VPC (p. 453)
- Instance Types Available Only in a VPC (p. 455)
- Amazon VPC Documentation (p. 455)
- Supported Platforms (p. 455)
- ClassicLink (p. 457)
- Migrating from a Windows Instance in EC2-Classic to a Windows Instance in a VPC (p. 465)

Benefits of Using a VPC

By launching your instances into a VPC instead of EC2-Classic, you gain the ability to:

- Assign static private IP addresses to your instances that persist across starts and stops
- Assign multiple IP addresses to your instances
- Define network interfaces, and attach one or more network interfaces to your instances
- Change security group membership for your instances while they're running
- Control the outbound traffic from your instances (egress filtering) in addition to controlling the inbound traffic to them (ingress filtering)
- Add an additional layer of access control to your instances in the form of network access control lists (ACL)
- Run your instances on single-tenant hardware

Differences Between EC2-Classic and EC2-VPC

The following table summarizes the differences between instances launched in EC2-Classic, instances launched in a default VPC, and instances launched in a nondefault VPC.

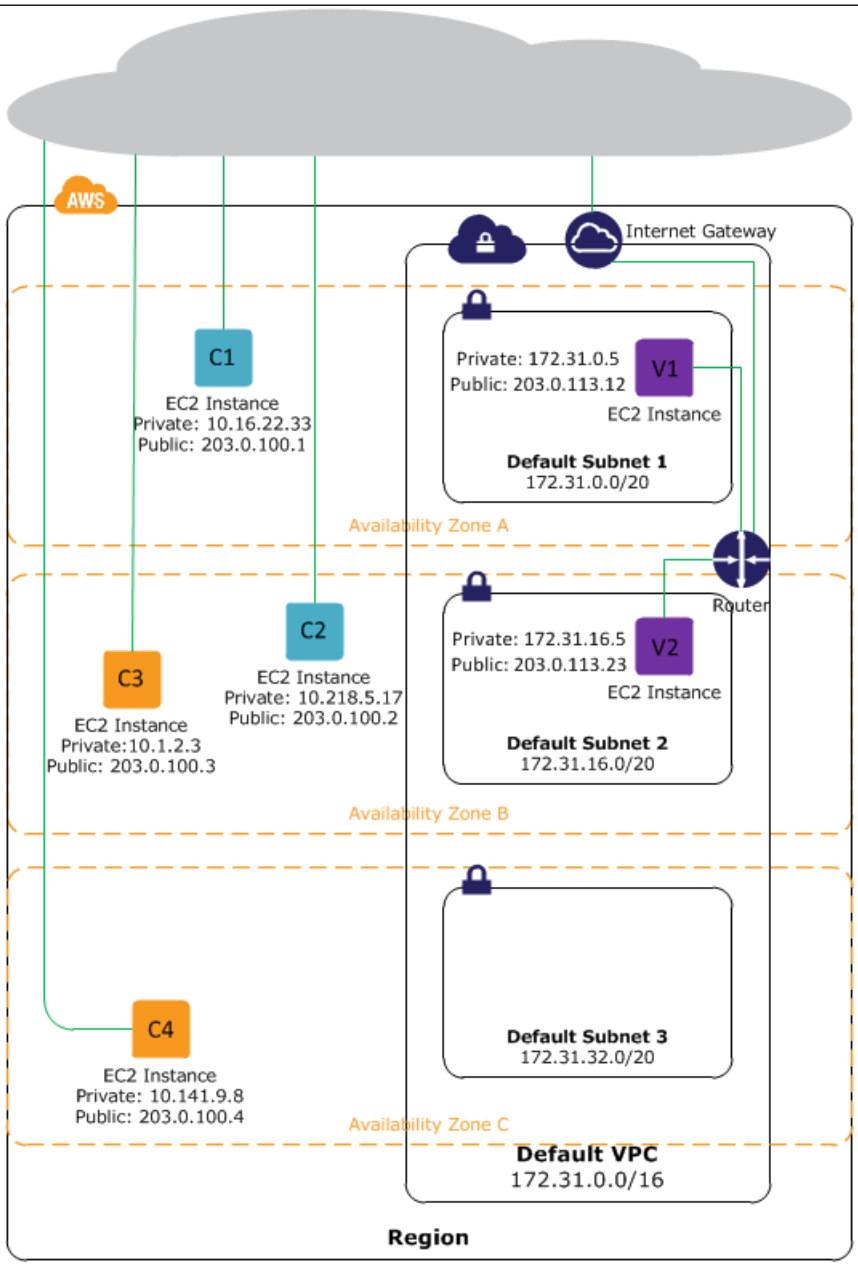
Characteristic	EC2-Classic	Default VPC	Nondefault VPC
Public IP address (from Amazon's public IP address pool)	Your instance receives a public IP address.	Your instance launched in a default subnet receives a public IP address by default, unless you specify otherwise during launch, or you modify the subnet's public IP address attribute.	Your instance doesn't receive a public IP address by default, unless you specify otherwise during launch, or you modify the subnet's public IP address attribute.
Private IP address	Your instance receives a private IP address from the EC2-Classic range each time it's started.	Your instance receives a static private IP address from the address range of your default VPC.	Your instance receives a static private IP address from the address range of your VPC.

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
Multiple private IP addresses	We select a single private IP address for your instance; multiple IP addresses are not supported.	You can assign multiple private IP addresses to your instance.	You can assign multiple private IP addresses to your instance.
Elastic IP address	An EIP is disassociated from your instance when you stop it.	An EIP remains associated with your instance when you stop it.	An EIP remains associated with your instance when you stop it.
DNS hostnames	DNS hostnames are enabled by default.	DNS hostnames are enabled by default.	DNS hostnames are disabled by default.
Security group	A security group can reference security groups that belong to other AWS accounts. You can create up to 500 security groups in each region.	A security group can reference security groups for your VPC only. You can create up to 100 security groups per VPC.	A security group can reference security groups for your VPC only. You can create up to 100 security groups per VPC.
Security group association	You can assign an unlimited number of security groups to an instance when you launch it. You can't change the security groups of your running instance. You can either modify the rules of the assigned security groups, or replace the instance with a new one (create an AMI from the instance, launch a new instance from this AMI with the security groups that you need, disassociate any Elastic IP address from the original instance and associate it with the new instance, and then terminate the original instance).	You can assign up to 5 security groups to an instance. You can assign security groups to your instance when you launch it and while it's running.	You can assign up to 5 security groups to an instance. You can assign security groups to your instance when you launch it and while it's running.
Security group rules	You can add rules for inbound traffic only. You can add up to 100 rules to a security group.	You can add rules for inbound and outbound traffic. You can add up to 50 rules to a security group.	You can add rules for inbound and outbound traffic. You can add up to 50 rules to a security group.
Tenancy	Your instance runs on shared hardware.	You can run your instance on shared hardware or single-tenant hardware.	You can run your instance on shared hardware or single-tenant hardware.

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
Accessing the Internet	Your instance can access the Internet. Your instance automatically receives a public IP address, and can access the Internet directly through the AWS network edge.	By default, your instance can access the Internet. Your instance receives a public IP address by default. An Internet gateway is attached to your default VPC, and your default subnet has a route to the Internet gateway.	By default, your instance cannot access the Internet. Your instance doesn't receive a public IP address by default. Your VPC may have an Internet gateway, depending on how it was created.

The following diagram shows instances in each platform. Note the following:

- Instances C1, C2, C3, and C4 are in the EC2-Classic platform. C1 and C2 were launched by one account, and C3 and C4 were launched by a different account. These instances can communicate with each other, can access the Internet directly.
- Instances V1 and V2 are in different subnets in the same VPC in the EC2-VPC platform. They were launched by the account that owns the VPC; no other account can launch instances in this VPC. These instances can communicate with each other and can access instances in EC2-Classic and the Internet through the Internet gateway.



Sharing and Accessing Resources Between EC2-Classic and EC2-VPC

Some resources and features in your AWS account can be shared or accessed between the EC2-Classic and EC2-VPC platforms, for example, through ClassicLink. For more information about ClassicLink, see ClassicLink (p. 457).

If your account supports EC2-Classic, you might have set up resources for use in EC2-Classic. If you want to migrate from EC2-Classic to a VPC, you must recreate those resources in your VPC. For more information about migrating from EC2-Classic to a VPC, see Migrating from a Windows Instance in EC2-Classic to a Windows Instance in a VPC (p. 465).

The following resources can be shared or accessed between EC2-Classic and a VPC.

Resource	Notes
AMI	
Bundle task	
EBS volume	
Elastic IP address	You can migrate an Elastic IP address from EC2-Classic to EC2-VPC. You can't migrate an Elastic IP address that was originally allocated for use in a VPC to EC2-Classic. For more information, see Migrating an Elastic IP Address from EC2-Classic to EC2-VPC (p. 487).
Instance	An EC2-Classic instance can communicate with instances in a VPC using public IP addresses, or you can use ClassicLink to enable communication over private IP addresses. You can't migrate an instance from EC2-Classic to a VPC. However, you can migrate your application from an instance in EC2-Classic to an instance in a VPC. For more information, see Migrating from a Windows Instance in EC2-Classic to a Windows Instance in a VPC (p. 465).
Key pair	
Load balancer	If you're using ClassicLink, you can register a linked EC2-Classic instance with a load balancer in a VPC, provided that the VPC has a subnet in the same Availability Zone as the instance. You can't migrate a load balancer from EC2-Classic to a VPC. You can't register an instance in a VPC with a load balancer in EC2-Classic.
Placement group	
Reserved Instance	You can change the network platform for your Reserved Instances from EC2-Classic to EC2-VPC.
Security group	A linked EC2-Classic instance can use a VPC security groups through ClassicLink to control traffic to and from the VPC. VPC instances can't use EC2-Classic security groups. You can't migrate a security group from EC2-Classic to a VPC. You can copy rules from a security group in EC2-Classic to a security group in a VPC. For more information, see Creating a Security Group (p. 402).
Snapshot	

The following resources can't be shared or moved between EC2-Classic and a VPC:

- Spot instances

Instance Types Available Only in a VPC

Instances of the following instance types are not supported in EC2-Classic and must be launched in a VPC:

- C4
- M4
- T2

If your account supports EC2-Classic but you have not created a nondefault VPC, you can do one of the following to launch a VPC-only instance:

- Create a nondefault VPC and launch your VPC-only instance into it by specifying a subnet ID or a network interface ID in the request. Note that you must create a nondefault VPC if you do not have a default VPC and you are using the AWS CLI, Amazon EC2 API, or Amazon EC2 CLI to launch a VPC-only instance. For more information, see Create a Virtual Private Cloud (VPC) (p. 17).
- Launch your VPC-only instance using the Amazon EC2 console. The Amazon EC2 console creates a nondefault VPC in your account and launches the instance into the subnet in the first Availability Zone. Note that the console creates the VPC with the following attributes:
  - One subnet in each Availability Zone, with the public IP addressing attribute set to `true` so that instances receive a public IP address. For more information, see IP Addressing in Your VPC in the *Amazon VPC User Guide*.
  - An Internet gateway, and a main route table that routes traffic in the VPC to the Internet gateway. This enables the instances you launch in the VPC to communicate over the Internet. For more information, see Internet Gateways in the *Amazon VPC User Guide*.
  - A default security group for the VPC and a default network ACL that is associated with each subnet. For more information, see Security in Your VPC in the *Amazon VPC User Guide*.

If you have other resources in EC2-Classic, you can take steps to migrate them to EC2-VPC. For more information, see Migrating from a Windows Instance in EC2-Classic to a Windows Instance in a VPC (p. 465).

Amazon VPC Documentation

For more information about Amazon VPC, see the following documentation.

Guide	Description
Amazon VPC Getting Started Guide	Provides a hands-on introduction to Amazon VPC.
Amazon VPC User Guide	Provides detailed information about how to use Amazon VPC.
Amazon VPC Network Administrator Guide	Helps network administrators configure your customer gateway.

Supported Platforms

Amazon EC2 supports the following platforms. Your AWS account is capable of launching instances either into both platforms or only into EC2-VPC, on a region by region basis.

Platform	Introduced In	Description
EC2-Classic	The original release of Amazon EC2	Your instances run in a single, flat network that you share with other customers.
EC2-VPC	The original release of Amazon VPC	Your instances run in a virtual private cloud (VPC) that's logically isolated to your AWS account.

For more information about the availability of either platform in your account, see Availability in the *Amazon VPC User Guide*. For more information about the differences between EC2-Classic and EC2-VPC, see Differences Between EC2-Classic and EC2-VPC (p. 450).

Supported Platforms in the Amazon EC2 Console

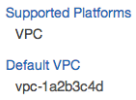
The Amazon EC2 console indicates which platforms you can launch instances into for the selected region, and whether you have a default VPC in that region.

Verify that the region you'll use is selected in the navigation bar. On the Amazon EC2 console dashboard, look for **Supported Platforms** under **Account Attributes**. If there are two values, `EC2` and `VPC`, you can launch instances into either platform. If there is one value, `VPC`, you can launch instances only into EC2-VPC.

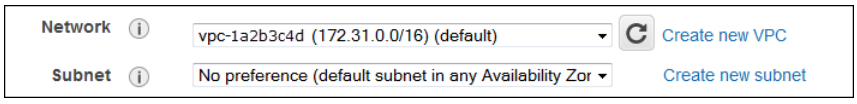
If you can launch instances only into EC2-VPC, we create a default VPC for you. Then, when you launch an instance, we launch it into your default VPC, unless you create a nondefault VPC and specify it when you launch the instance.

EC2-VPC

The dashboard displays the following under **Account Attributes** to indicate that the account supports only the EC2-VPC platform, and has a default VPC with the identifier `vpc-1a2b3c4d`.



If your account supports only EC2-VPC, you can select a VPC from the **Network** list, and a subnet from the **Subnet** list when you launch an instance using the launch wizard.



EC2-Classic, EC2-VPC

The dashboard displays the following under **Account Attributes** to indicate that the account supports both the EC2-Classic and EC2-VPC platforms.

If your account supports EC2-Classic and EC2-VPC, you can launch into EC2-Classic using the launch wizard by selecting **Launch into EC2-Classic** from the **Network** list. To launch into a VPC, you can select a VPC from the **Network** list, and a subnet from the **Subnet** list.

Related Topic

For more information about how you can tell which platforms you can launch instances into, see Detecting Your Supported Platforms in the *Amazon VPC User Guide*.

ClassicLink

ClassicLink allows you to link your EC2-Classic instance to a VPC in your account, within the same region. This allows you to associate the VPC security groups with the EC2-Classic instance, enabling communication between your EC2-Classic instance and instances in your VPC using private IP addresses. ClassicLink removes the need to make use of public IP addresses or Elastic IP addresses to enable communication between instances in these platforms. For more information about private and public IP addresses, see IP Addressing in Your VPC.

ClassicLink is available to all users with accounts that support the EC2-Classic platform, and can be used with any EC2-Classic instance. To find out which platform your account supports, see Supported Platforms (p. 455). For more information about the benefits of using a VPC, see Amazon EC2 and Amazon Virtual Private Cloud (p. 449). For more information about migrating your resources to a VPC, see Migrating from a Windows Instance in EC2-Classic to a Windows Instance in a VPC (p. 465).

There is no additional charge for using ClassicLink. Standard charges for data transfer and instance hour usage apply.

Topics

- ClassicLink Basics (p. 457)
- ClassicLink Limitations (p. 459)
- Working with ClassicLink (p. 460)
- API and CLI Overview (p. 463)
- Example: ClassicLink Security Group Configuration for a Three-Tier Web Application (p. 464)

ClassicLink Basics

There are two steps to linking an EC2-Classic instance to a VPC using ClassicLink. First, you must enable the VPC for ClassicLink. By default, all VPCs in your account are not enabled for ClassicLink, to maintain their isolation. After you've enabled the VPC for ClassicLink, you can then link any running EC2-Classic instance in the same region in your account to that VPC. Linking your instance includes selecting security groups from the VPC to associate with your EC2-Classic instance. After you've linked the instance, it can communicate with instances in your VPC using their private IP addresses, provided the VPC security groups allow it. Your EC2-Classic instance does not lose its private IP address when linked to the VPC.

> **Note**
> Linking your instance to a VPC is sometimes referred to as *attaching* your instance.

A linked EC2-Classic instance can communicate with instances in a VPC, but it does not form part of the VPC. If you list your instances and filter by VPC, for example, through the `DescribeInstances` API request, or by using the **Instances** screen in the Amazon EC2 console, the results do not return any EC2-Classic instances that are linked to the VPC. For more information about viewing your linked EC2-Classic instances, see Viewing Your ClassicLink-Enabled VPCs and Linked EC2-Classic Instances (p. 462).

If you no longer require a ClassicLink connection between your instance and the VPC, you can unlink the EC2-Classic instance from the VPC. This disassociates the VPC's security groups from the EC2-Classic instance. A linked EC2-Classic instance is automatically unlinked from a VPC when it's stopped. After you've unlinked all linked EC2-Classic instances from the VPC, you can disable ClassicLink for the VPC.

Using Other AWS Services in Your VPC With ClassicLink

Linked EC2-Classic instances can access the following AWS services in the VPC: Amazon Redshift, Amazon ElastiCache, Elastic Load Balancing, and Amazon RDS. However, instances in the VPC cannot access the AWS services provisioned by the EC2-Classic platform using ClassicLink.

If you use Elastic Load Balancing in your VPC, you can register your linked EC2-Classic instance with the load balancer, provided that the instance is in an Availability Zone in which your VPC has a subnet. If you terminate the linked EC2-Classic instance, the load balancer deregisters the instance. For more information about working with load balancers in a VPC, see Elastic Load Balancing in Amazon VPC in the *Elastic Load Balancing Developer Guide*.

If you use Auto Scaling, you can create an Auto Scaling group with instances that are automatically linked to a specified ClassicLink-enabled VPC at launch. For more information, see Linking EC2-Classic Instances to a VPC in the *Auto Scaling Developer Guide*.

If you use Amazon RDS instances or Amazon Redshift clusters in your VPC, and they are publicly accessible (accessible from the Internet), the endpoint you use to address those resources from a linked EC2-Classic instance resolves to a public IP address. If those resources are not publicly accessible, the endpoint resolves to a private IP address. To address a publicly accessible RDS instance or Redshift cluster over private IP using ClassicLink, you must use their private IP address or private DNS hostname.

If you use a private DNS hostname or a private IP address to address an RDS instance, the linked EC2-Classic instance cannot use the failover support available for Multi-AZ deployments.

You can use the Amazon EC2 console to find the private IP addresses of your Amazon Redshift, Amazon ElastiCache, or Amazon RDS resources.

To locate the private IP addresses of AWS resources in your VPC

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Network Interfaces**.
3. Check the descriptions of the network interfaces in the **Description** column. A network interface that's used by Amazon Redshift, Amazon ElastiCache, or Amazon RDS will have the name of the service in the description. For example, a network interface that's attached to an Amazon RDS instance will have the following description: `RDSNetworkInterface`.
4. Select the required network interface.
5. In the details pane, get the private IP address from the **Primary private IP** field.

Controlling the Use of ClassicLink

By default, IAM users do not have permission to work with ClassicLink. You can create an IAM policy that grants users permissions to enable or disable a VPC for ClassicLink, link or unlink an instance to a ClassicLink-enabled VPC, and to view ClassicLink-enabled VPCs and linked EC2-Classic instances. For more information about IAM policies for Amazon EC2, see IAM Policies for Amazon EC2 (p. 408).

For more information about policies for working with ClassicLink, see the following example: 6. Allow users to work with ClassicLink (p. 431).

Security Groups in ClassicLink

Linking your EC2-Classic instance to a VPC does not affect your EC2-Classic security groups. They continue to control all traffic to and from the instance. This excludes traffic to and from instances in the VPC, which is controlled by the VPC security groups that you associated with the EC2-Classic instance. EC2-Classic instances that are linked to the same VPC cannot communicate with each other through the VPC; regardless of whether they are associated with the same VPC security group. Communication between EC2-Classic instances is controlled by the EC2-Classic security groups associated with those

instances. For an example of a security group configuration, see Example: ClassicLink Security Group Configuration for a Three-Tier Web Application (p. 464).

After you've linked your instance to a VPC, you cannot change which VPC security groups are associated with the instance. To associate different security groups with your instance, you must first unlink the instance, and then link it to the VPC again, choosing the required security groups.

Routing for ClassicLink

When you enable a VPC for ClassicLink, a static route is added to all of the VPC route tables with a destination of `10.0.0.0/8` and a target of `local`. This allows communication between instances in the VPC and any EC2-Classic instances that are then linked to the VPC. If you add a custom route table to a ClassicLink-enabled VPC, a static route is automatically added with a destination of `10.0.0.0/8` and a target of `local`. When you disable ClassicLink for a VPC, this route is automatically deleted in all of the VPC route tables.

VPCs that are in the `10.0.0.0/16` and `10.1.0.0/16` IP address ranges can be enabled for ClassicLink only if they do not have any existing static routes in route tables in the `10.0.0.0/8` IP address range, excluding the local routes that were automatically added when the VPC was created. Similarly, if you've enabled a VPC for ClassicLink, you may not be able to add any more specific routes to your route tables within the `10.0.0.0/8` IP address range.

> **Important**
> If your VPC's CIDR block is a publicly routable IP address range, consider the security implications before you link an EC2-Classic instance to your VPC. For example, if your linked EC2-Classic instance receives an incoming Denial of Service (DoS) request flood attack from a source IP address that falls within the VPC's IP address range, the response traffic is sent into your VPC. We strongly recommend that you create your VPC using a private IP address range as specified in RFC 1918.

For more information about route tables and routing in your VPC, see Route Tables in the *Amazon VPC User Guide*.

ClassicLink Limitations

To use the ClassicLink feature, you need to be aware of the following limitations:

- You can link an EC2-Classic instance to only one VPC at a time.
- If you stop your linked EC2-Classic instance, it's automatically unlinked from the VPC, and the VPC security groups are no longer associated with the instance. You can link your instance to the VPC again after you've restarted it.
- You cannot link an EC2-Classic instance to a VPC that's in a different region, or a different AWS account.
- VPCs configured for dedicated hardware tenancy cannot be enabled for ClassicLink. Contact AWS support to request that your dedicated tenancy VPC be allowed to be enabled for ClassicLink.

  > **Important**
  > EC2-Classic instances are run on shared hardware. If you've set the tenancy of your VPC to `dedicated` because of regulatory or security requirements, then linking an EC2-Classic instance to your VPC may not conform to those requirements, as you will be allowing a shared tenancy resource to address your isolated resources directly using private IP addresses. If you want to enable your dedicated VPC for ClassicLink, provide a detailed motivation in your request to AWS support.

- VPCs with routes that conflict with the EC2-Classic private IP address range of `10/8` cannot be enabled for ClassicLink. This does not include VPCs with `10.0.0.0/16` and `10.1.0.0/16` IP address ranges that already have local routes in their route tables. For more information, see Routing for ClassicLink (p. 459).

- You cannot associate a VPC Elastic IP address with a linked EC2-Classic instance.
- If you use a public DNS hostname to address an instance in a VPC from a linked EC2-Classic instance, the hostname does not resolve to the instance's private IP address. Instead, the public DNS hostname resolves to the public IP address. The same applies if you use a public DNS hostname to address a linked EC2-Classic instance from an instance in a VPC.
- You can link a running Spot instance to a VPC. To indicate in a Spot instance request that the instance should be linked to a VPC when the request is fulfilled, you must use the launch wizard in the Amazon EC2 console.
- ClassicLink does not support transitive relationships out of the VPC. Your linked EC2-Classic instance will not have access to any VPN connection, VPC peering connection, VPC endpoint, or Internet gateway associated with the VPC. Similarly, resources on the other side of a VPN connection, a VPC peering connection, or an Internet gateway will not have access to a linked EC2-Classic instance.
- You cannot use ClassicLink to link a VPC instance to a different VPC, or to a EC2-Classic resource. To establish a private connection between VPCs, you can use a VPC peering connection. For more information, see VPC Peering in the *Amazon VPC User Guide*.
- If you link your EC2-Classic instance to a VPC in the `172.16.0.0/16` range, and you have a DNS server running on the `172.16.0.23/32` IP address within the VPC, then your linked EC2-Classic instance will not be able to access the VPC DNS server. To work around this issue, run your DNS server on a different IP address within the VPC.

Working with ClassicLink

You can use the Amazon EC2 and Amazon VPC consoles to work with the ClassicLink feature. You can enable or disable a VPC for ClassicLink, and link and unlink EC2-Classic instances to a VPC.

> **Note**
> The ClassicLink features are only visible in the consoles for accounts and regions that support EC2-Classic.

Topics

- Enabling a VPC for ClassicLink (p. 460)
- Linking an Instance to a VPC (p. 461)
- Creating a VPC with ClassicLink Enabled (p. 461)
- Linking an EC2-Classic Instance to a VPC at Launch (p. 461)
- Viewing Your ClassicLink-Enabled VPCs and Linked EC2-Classic Instances (p. 462)
- Unlink a EC2-Classic Instance from a VPC (p. 462)
- Disable ClassicLink for a VPC (p. 462)

Enabling a VPC for ClassicLink

To link an EC2-Classic instance to a VPC, you must first enable the VPC for ClassicLink. You cannot enable a VPC for ClassicLink if the VPC has routing that conflicts with the EC2-Classic private IP address range. For more information, see Routing for ClassicLink (p. 459).

To enable a VPC for ClassicLink

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, click **Your VPCs**.
3. Select a VPC, and then select **Enable ClassicLink** from the **Actions** list.
4. In the confirmation dialog box, click **Yes, Enable**.

Linking an Instance to a VPC

After you've enabled a VPC for ClassicLink, you can link an EC2-Classic instance to it.

> **Note**
> You can only link a running EC2-Classic instance to a VPC. You cannot link an instance that's in the `stopped` state.

To link an instance to a VPC

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Instances**.
3. Select the running EC2-Classic instance, click **Actions**, select **ClassicLink**, and then click **Link to VPC**. You can select more than one instance to link to the same VPC.
4. In the dialog box that displays, select a VPC from the list. Only VPCs that have been enabled for ClassicLink are displayed.
5. Select one or more of the VPC security groups to associate with your instance. When you are done, click **Link to VPC**.

Creating a VPC with ClassicLink Enabled

You can create a new VPC and immediately enable it for ClassicLink by using the VPC wizard in the Amazon VPC console.

To create a VPC with ClassicLink enabled

1. Open the Amazon VPC console.
2. From the Amazon VPC dashboard, click **Start VPC Wizard**.
3. Choose one of the VPC configuration options and click **Select**.
4. On the next page of the wizard, select **Yes** in the **Enable ClassicLink** field. Complete the rest of the steps in the wizard to create your VPC. For more information about using the VPC wizard, see Scenarios for Amazon VPC in the *Amazon VPC User Guide*.

Linking an EC2-Classic Instance to a VPC at Launch

You can use the launch wizard in the Amazon EC2 console to launch an EC2-Classic instance and immediately link it to a ClassicLink-enabled VPC.

To link an instance to a VPC at launch

1. Open the Amazon EC2 console.
2. From the Amazon EC2 dashboard, click **Launch Instance**.
3. Select an AMI, and then choose an instance type. On the **Configure Instance Details** page, ensure that you select **Launch into EC2-Classic** from the **Network** list.

   > **Note**
   > Some instance types, such as T2 instance types, can only be launched into a VPC. Ensure that you select an instance type that can be launched into EC2-Classic.

4. In the **Link to VPC (ClassicLink)** section, select a VPC from the **Link to VPC** list. Only ClassicLink-enabled VPCs are displayed. Select the security groups from the VPC to associate with the instance. Complete the other configuration options on the page, and then complete the rest of the steps in the wizard to launch your instance. For more information about using the launch wizard, see Launching Your Instance from an AMI (p. 207).

Viewing Your ClassicLink-Enabled VPCs and Linked EC2-Classic Instances

You can view all of your ClassicLink-enabled VPCs in the Amazon VPC console, and your linked EC2-Classic instances in the Amazon EC2 console.

To view your ClassicLink-enabled VPCs

1. Open the Amazon VPC console.
2. In the navigation pane, click **Your VPCs**.
3. Select a VPC, and in the **Summary** tab, look for the **ClassicLink** field. A value of **Enabled** indicates that the VPC is enabled for ClassicLink.
4. Alternatively, look for the **ClassicLink** column, and view the value that's displayed for each VPC (**Enabled** or **Disabled**). If the column is not visible, click **Edit Table Columns** (the gear-shaped icon), select the **ClassicLink** attribute, and then click **Close**.

To view your linked EC2-Classic instances

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**.
3. Select an EC2-Classic instance, and in the **Description** tab, look for the **ClassicLink** field. If the instance is linked to a VPC, the field displays the ID of the VPC to which the instance is linked. If the instance is not linked to any VPC, the field displays **Unlinked**.
4. Alternatively, you can filter your instances to display only linked EC2-Classic instances for a specific VPC or security group. In the search bar, start typing `ClassicLink`, select the relevant ClassicLink resource attribute, and then select the security group ID or the VPC ID.

Unlink a EC2-Classic Instance from a VPC

If you no longer require a ClassicLink connection between your EC2-Classic instance and your VPC, you can unlink the instance from the VPC. Unlinking the instance disassociates the VPC security groups from the instance.

Note
A stopped instance is automatically unlinked from a VPC.

To unlink an instance from a VPC

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**, and select your instance.
3. In the **Actions** list, select **ClassicLink**, and then **Unlink Instance**. You can select more than one instance to unlink from the same VPC.
4. Click **Yes** in the confirmation dialog box.

Disable ClassicLink for a VPC

If you no longer require a connection between EC2-Classic instances and your VPC, you can disable ClassicLink on the VPC. You must first unlink all linked EC2-Classic instances that are linked to the VPC.

To disable ClassicLink for a VPC

1. Open the VPC console.
2. In the navigation pane, click **Your VPCs**.
3. Select your VPC, then select **Disable ClassicLink** from the **Actions** list.

4. In the confirmation dialog box, click **Yes, Disable**.

API and CLI Overview

You can perform the tasks described on this page using the command line or the Query API. For more information about the command line interfaces and a list of available API actions, see Accessing Amazon EC2 (p. 3).

Enable a VPC for ClassicLink

- enable-vpc-classic-link (AWS CLI)
- ec2-enable-vpc-classic-link (Amazon EC2 CLI)
- Enable-EC2VpcClassicLink (AWS Tools for Windows PowerShell)
- EnableVpcClassicLink(Amazon EC2 Query API)

Link (attach) an EC2-Classic instance to a VPC

- attach-classic-link-vpc (AWS CLI)
- ec2-attach-classic-link-vpc (Amazon EC2 CLI)
- Add-EC2ClassicLinkVpc (AWS Tools for Windows PowerShell)
- AttachClassicLinkVpc(Amazon EC2 Query API)

Unlink (detach) an EC2-Classic instance from a VPC

- detach-classic-link-vpc (AWS CLI)
- ec2-detach-classic-link-vpc (Amazon EC2 CLI)
- Dismount-EC2ClassicLinkVpc (AWS Tools for Windows PowerShell)
- DetachClassicLinkVpc(Amazon EC2 Query API)

Disable ClassicLink for a VPC

- disable-vpc-classic-link (AWS CLI)
- ec2-disable-vpc-classic-link (Amazon EC2 CLI)
- Disable-EC2VpcClassicLink (AWS Tools for Windows PowerShell)
- DisableVpcClassicLink(Amazon EC2 Query API)

Describe the ClassicLink status of VPCs

- describe-vpc-classic-link (AWS CLI)
- ec2-describe-vpc-classic-link (Amazon EC2 CLI)
- Get-EC2VpcClassicLink (AWS Tools for Windows PowerShell)
- DescribeVpcClassicLink(Amazon EC2 Query API)

Describe linked EC2-Classic instances

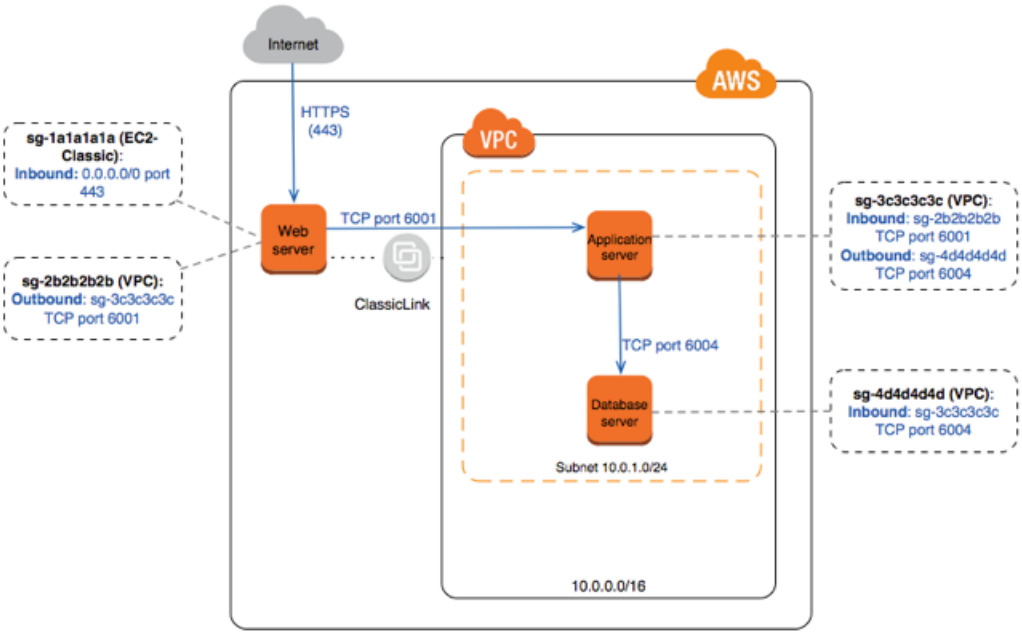
- describe-classic-link-instances (AWS CLI)
- ec2-describe-classic-link-instances (Amazon EC2 CLI)
- Get-EC2ClassicLinkInstance (AWS Tools for Windows PowerShell)
- DescribeClassicLinkInstances(Amazon EC2 Query API)

Example: ClassicLink Security Group Configuration for a Three-Tier Web Application

In this example, you have an application with three instances: a public-facing web server, an application server, and a database server. Your web server accepts HTTPS traffic from the Internet, and then communicates with your application server over TCP port 6001. Your application server then communicates with your database server over TCP port 6004. You're in the process of migrating your entire application to a VPC in your account. You've already migrated your application server and your database server to your VPC. Your web server is still in EC2-Classic and linked to your VPC via ClassicLink.

You want a security group configuration that allows traffic to flow only between these instances. You have four security groups: two for your web server (`sg-1a1a1a1a` and `sg-2b2b2b2b`), one for your application server (`sg-3c3c3c3c`), and one for your database server (`sg-4d4d4d4d`).

The following diagram displays the architecture of your instances, and their security group configuration.



Security Groups for Your Web Server (`sg-1a1a1a1a` and `sg-2b2b2b2b`)

You have one security group in EC2-Classic, and the other in your VPC. You associated the the VPC security group with your web server instance when you linked the instance to your VPC via ClassicLink. The VPC security group enables you to control the outbound traffic from your web server to your application server.

The following are the security group rules for the EC2-Classic security group (`sg-1a1a1a1a`).

Inbound			
Source	**Type**	**Port Range**	**Comments**
0.0.0.0/0	HTTPS	443	Allows Internet traffic to reach your web server.

The following are the security group rules for the VPC security group (`sg-2b2b2b2b`).

Outbound			
Destination	**Type**	**Port Range**	**Comments**
sg-3c3c3c3c	TCP	6001	Allows outbound traffic from your web server to your application server in your VPC (or to any other instance associated with `sg-3c3c3c3c`).

Security Group for Your Application Server (`sg-3c3c3c3c`)

The following are the security group rules for the VPC security group that's associated with your application server.

Inbound			
Source	**Type**	**Port Range**	**Comments**
sg-2b2b2b2b	TCP	6001	Allows the specified type of traffic from your web server (or any other instance associated with `sg-2b2b2b2b`) to reach your application server.
Outbound			
Destination	**Type**	**Port Range**	**Comments**
sg-4d4d4d4d	TCP	6004	Allows outbound traffic from the application server to the database server (or to any other instance associated with `sg-4d4d4d4d`).

Security Group for Your Database Server (`sg-4d4d4d4d`)

The following are the security group rules for the VPC security group that's associated with your database server.

Inbound			
Source	**Type**	**Port Range**	**Comments**
sg-3c3c3c3c	TCP	6004	Allows the specified type of traffic from your application server (or any other instance associated with `sg-3c3c3c3c`) to reach your database server.

Migrating from a Windows Instance in EC2-Classic to a Windows Instance in a VPC

Your AWS account might support both EC2-Classic and EC2-VPC, depending on when you created your account and which regions you've used. For more information, and to find out which platform your account supports, see Supported Platforms (p. 455). For more information about the benefits of using a VPC, and

the differences between EC2-Classic and EC2-VPC, see Amazon EC2 and Amazon Virtual Private Cloud (p. 449).

You create and use resources in your AWS account. Some resources and features, such as enhanced networking and T2 instances, can be used only in a VPC. Some resources can be shared between EC2-Classic and a VPC, while some can't. For more information, see Sharing and Accessing Resources Between EC2-Classic and EC2-VPC (p. 453).

If your account supports EC2-Classic, you might have set up resources for use in EC2-Classic. If you want to migrate from EC2-Classic to a VPC, you must recreate those resources in your VPC.

There are two ways of migrating to a VPC. You can do a full migration, or you can do an incremental migration over time. The method you choose depends on the size and complexity of your application in EC2-Classic. For example, if your application consists of one or two instances running a static website, and you can afford a short period of downtime, you can do a full migration. If you have a multi-tier application with processes that cannot be interrupted, you can do an incremental migration using ClassicLink. This allows you to transfer functionality one component at a time until your application is running fully in your VPC.

If you need to migrate a Linux instance, see Migrating a Linux Instance from EC2-Classic to a VPC in the *Amazon EC2 User Guide for Linux Instances*.

Contents

- Full Migration to a VPC (p. 466)
- Incremental Migration to a VPC Using ClassicLink (p. 471)

Full Migration to a VPC

Complete the following tasks to fully migrate your application from EC2-Classic to a VPC.

Tasks

- Step 1: Create a VPC (p. 466)
- Step 2: Configure Your Security Group (p. 467)
- Step 3: Create an AMI from Your EC2-Classic Instance (p. 467)
- Step 4: Launch an Instance Into Your VPC (p. 468)
- Example: Migrating a Simple Web Application (p. 469)

Step 1: Create a VPC

To start using a VPC, ensure that you have one in your account. You can create one using one of these methods:

- Use a new, EC2-VPC-only AWS account. Your EC2-VPC-only account comes with a default VPC in each region, which is ready for you to use. Instances that you launch are by default launched into this VPC, unless you specify otherwise. For more information about your default VPC, see Your Default VPC and Subnets. Use this option if you'd prefer not to set up a VPC yourself, or if you do not need specific requirements for your VPC configuration.
- In your existing AWS account, open the Amazon VPC console and use the VPC wizard to create a new VPC. For more information, see Scenarios for Amazon VPC. Use this option if you want to set up a VPC quickly in your existing EC2-Classic account, using one of the available configuration sets in the wizard. You'll specify this VPC each time you launch an instance.
- In your existing AWS account, open the Amazon VPC console and set up the components of a VPC according to your requirements. For more information, see Your VPC and Subnets. Use this option if you have specific requirements for your VPC, such as a particular number of subnets. You'll specify this VPC each time you launch an instance.

Note
T2 instance types must be launched into a VPC. If you do not have any VPCs in your EC2-Classic account, and you use the launch wizard in the Amazon EC2 console to launch a T2 instance, the wizard creates a nondefault VPC for you. For more information about T2 instance types, see T2 Instances (p. 99). Your T2 instance will not be able to communicate with your EC2-Classic instances using private IP addresses. Consider migrating your existing instances to the same VPC using the methods outlined in this topic.

Step 2: Configure Your Security Group

You cannot use the same security groups between EC2-Classic and a VPC. However, if you want your instances in your VPC to have the same security group rules as your EC2-Classic instances, you can use the Amazon EC2 console to copy your existing EC2-Classic security group rules to a new VPC security group.

Important
You can only copy security group rules to a new security group in the same AWS account in the same region. If you've created a new AWS account, you cannot use this method to copy your existing security group rules to your new account. You'll have to create a new security group, and add the rules yourself. For more information about creating a new security group, see Amazon EC2 Security Groups for Windows Instances (p. 398).

To copy your security group rules to a new security group

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group that's associated with your EC2-Classic instance, then choose **Actions** and select **Copy to new**.
4. In the **Create Security Group** dialog box, specify a name and description for your new security group. Select your VPC from the **VPC** list.
5. The **Inbound** tab is populated with the rules from your EC2-Classic security group. You can modify the rules as required. In the **Outbound** tab, a rule that allows all outbound traffic has automatically been created for you. For more information about modifying security group rules, see Amazon EC2 Security Groups for Windows Instances (p. 398).

   Note
   If you've defined a rule in your EC2-Classic security group that references another security group, you will not be able to use the same rule in your VPC security group. Modify the rule to reference a security group in the same VPC.
6. Choose **Create**.

Step 3: Create an AMI from Your EC2-Classic Instance

An AMI is a template for launching your instance. You can create your own AMI based on an existing EC2-Classic instance, then use that AMI to launch instances into your VPC.

The method you use to create your AMI depends on the root device type of your instance, and the operating system platform on which your instance runs. To find out the root device type of your instance, go to the **Instances** page, select your instance, and look at the information in the **Root device type** field in the **Description** tab. If the value is `ebs`, then your instance is EBS-backed. If the value is `instance-store`, then your instance is instance store-backed. You can also use the describe-instances AWS CLI command to find out the root device type.

The following table provides options for you to create your AMI based on the root device type of your instance, and the software platform.

Instance Root Device Type	Action
EBS	Create an EBS-backed AMI from your instance. For more information, see Creating an Amazon EBS-Backed Windows AMI (p. 68).
Instance store	Bundle your instance, and then create an instance store-backed AMI from the manifest that's created during bundling. For more information, see Creating an Instance Store-Backed Windows AMI (p. 70).

(Optional) Store Your Data on Amazon EBS Volumes

You can create an Amazon EBS volume and use it to back up and store the data on your instance—like you would use a physical hard drive. Amazon EBS volumes can be attached and detached from any instance in the same Availability Zone. You can detach a volume from your instance in EC2-Classic, and attach it to a new instance that you launch into your VPC in the same Availability Zone.

For more information about Amazon EBS volumes, see the following topics:

- Amazon EBS Volumes (p. 518)
- Creating an Amazon EBS Volume (p. 525)
- Attaching an Amazon EBS Volume to an Instance (p. 529)

To back up the data on your Amazon EBS volume, you can take periodic snapshots of your volume. If you need to, you can restore an Amazon EBS volume from your snapshot. For more information about Amazon EBS snapshots, see the following topics:

- Amazon EBS Snapshots (p. 549)
- Creating an Amazon EBS Snapshot (p. 550)
- Restoring an Amazon EBS Volume from a Snapshot (p. 527)

Step 4: Launch an Instance Into Your VPC

After you've created an AMI, you can launch an instance into your VPC. The instance will have the same data and configurations as your existing EC2-Classic instance.

You can either launch your instance into a VPC that you've created in your existing account, or into a new, VPC-only AWS account.

Using Your Existing EC2-Classic Account

You can use the Amazon EC2 launch wizard to launch an instance into your VPC.

To launch an instance into your VPC

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the dashboard, choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image** page, select the **My AMIs** category, and select the AMI you created.
4. On the **Choose an Instance Type** page, select the type of instance, and choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, select your VPC from the **Network** list. Select the required subnet from the **Subnet** list. Configure any other details you require, then go through the next pages of the wizard until you reach the **Configure Security Group** page.

6. Select **Select an existing group**, and select the security group you created earlier. Choose **Review and Launch**.
7. Review your instance details, then choose **Launch** to specify a key pair and launch your instance.

For more information about the parameters you can configure in each step of the wizard, see Launching an Instance (p. 207).

Using Your New, VPC-Only Account

To launch an instance in your new AWS account, you'll first have to share the AMI you created with your new account. You can then use the Amazon EC2 launch wizard to launch an instance into your default VPC.

To share an AMI with your new AWS account

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Switch to the account in which you created your AMI.
3. In the navigation pane, choose **AMIs**.
4. In the **Filter** list, ensure **Owned by me** is selected, then select your AMI.
5. In the **Permissions** tab, choose **Edit**. Enter the account number of your new AWS account, choose **Add Permission**, and then choose **Save**.

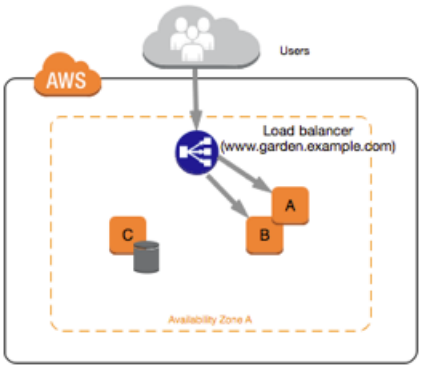
To launch an instance into your default VPC

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Switch to your new AWS account.
3. In the navigation pane, choose **AMIs**.
4. In the **Filter** list, select **Private images**. Select the AMI that you shared from your EC2-Classic account, then choose **Launch**.
5. On the **Choose an Instance Type** page, select the type of instance, and choose **Next: Configure Instance Details**.
6. On the **Configure Instance Details** page, your default VPC should be selected in the **Network** list. Configure any other details you require, then go through the next pages of the wizard until you reach the **Configure Security Group** page.
7. Select **Select an existing group**, and select the security group you created earlier. Choose **Review and Launch**.
8. Review your instance details, then choose **Launch** to specify a key pair and launch your instance.

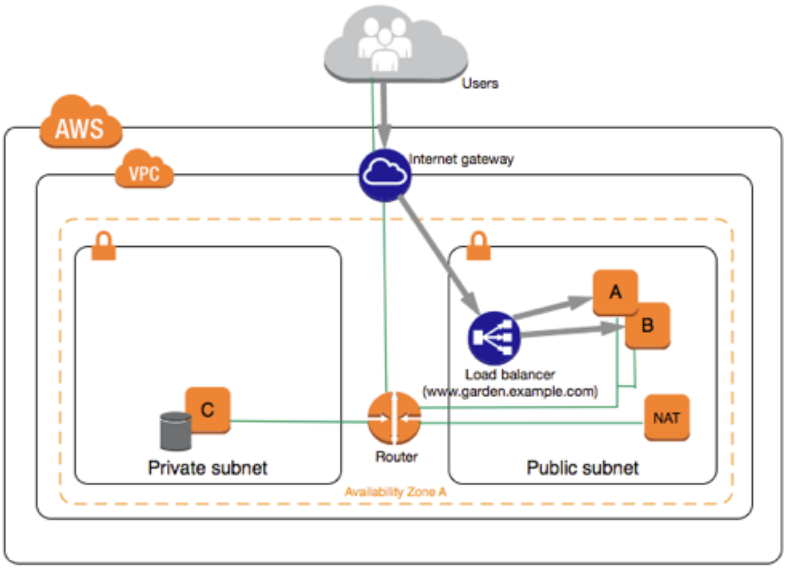
For more information about the parameters you can configure in each step of the wizard, see Launching an Instance (p. 207).

Example: Migrating a Simple Web Application

In this example, you use AWS to host your gardening website. To manage your website, you have three running instances in EC2-Classic. Instances A and B host your public-facing web application, and you use an Elastic Load Balancer to load balance the traffic between these instances. You've assigned Elastic IP addresses to instances A and B so that you have static IP addresses for configuration and administration tasks on those instances. Instance C holds your MySQL database for your website. You've registered the domain name `www.garden.example.com`, and you've used Amazon Route 53 to create a hosted zone with an alias record set that's associated with the DNS name of your load balancer.



The first part of migrating to a VPC is deciding what kind of VPC architecture will suit your needs. In this case, you've decided on the following: one public subnet for your web servers, and one private subnet for your database server. As your website grows, you can add more web servers and database servers to your subnets. By default, instances in the private subnet cannot access the Internet; however, you can enable Internet access through a Network Address Translation (NAT) instance in the public subnet. You may want to set up a NAT instance to support periodic updates and patches from the Internet for your database server. You'll migrate your Elastic IP addresses to EC2-VPC, and create an Elastic Load Balancer in your public subnet to load balance the traffic between your web servers.



To migrate your web application to a VPC, you can follow these steps:

- **Create a VPC**: In this case, you can use the VPC wizard in the Amazon VPC console to create your VPC and subnets. The second wizard configuration creates a VPC with one private and one public subnet, and launches and configures a NAT instance in your public subnet for you. For more information, see Scenario 2: VPC with Public and Private Subnets in the *Amazon VPC User Guide*.
- **Create AMIs from your instances**: Create an AMI from one of your web servers, and a second AMI from your database server. For more information, see Step 3: Create an AMI from Your EC2-Classic Instance (p. 467).
- **Configure your security groups**: In your EC2-Classic environment, you have one security group for your web servers, and another security group for your database server. You can use the Amazon EC2 console to copy the rules from each security group into new security groups for your VPC. For more information, see Step 2: Configure Your Security Group (p. 467).

Tip
Create the security groups that are referenced by other security groups first.

- **Launch an instance into your new VPC**: Launch replacement web servers into your public subnet, and launch your replacement database server into your private subnet. For more information, see Step 4: Launch an Instance Into Your VPC (p. 468).
- **Configure your NAT instance**: If you want to make use of your NAT instance to allow your database server to access the Internet, you'll have to create a security group for your NAT instance that allows HTTP and HTTPS traffic from your private subnet. For more information, see NAT Instances.
- **Configure your database**: When you created an AMI from your database server in EC2-Classic, all the configuration information that was stored in that instance was copied to the AMI. You may have to connect to your new database server and update the configuration details; for example, if you configured your database to grant full read, write, and modification permissions to your web servers in EC2-Classic, you'll have to update the configuration files to grant the same permissions to your new VPC web servers instead.
- **Configure your web servers**: Your web servers will have the same configuration settings as your instances in EC2-Classic. For example, if you configured your web servers to use the database in EC2-Classic, update your web servers' configuration settings to point to your new database instance.

  Note
  By default, instances launched into a nondefault subnet are not assigned a public IP address, unless you specify otherwise at launch. Your new database server may not have a public IP address. In this case, you can update your web servers' configuration file to use your new database server's private DNS name. Instances in the same VPC can communicate with each other via private IP address.

- **Migrate your Elastic IP addresses**: Disassociate your Elastic IP addresses from your web servers in EC2-Classic, and then migrate them to EC2-VPC. After you've migrated them, you can associate them with your new web servers in your VPC. For more information, see Migrating an Elastic IP Address from EC2-Classic to EC2-VPC (p. 487).
- **Create a new load balancer**: To continue using Elastic Load Balancing to load balance the traffic to your instances, make sure you understand the various ways you can configure your load balancer in VPC. For more information, see Elastic Load Balancing in Amazon VPC.
- **Update your DNS records**: After you've set up your load balancer in your public subnet, ensure that your `www.garden.example.com` domain points to your new load balancer. To do this, you'll need to update your DNS records and update your alias record set in Amazon Route 53. For more information about using Amazon Route 53, see Getting Started with Amazon Route 53.
- **Shut down your EC2-Classic resources**: After you've verified that your web application is working from within the VPC architecture, you can shut down your EC2-Classic resources to stop incurring charges for them. Terminate your EC2-Classic instances, and release your EC2-Classic Elastic IP addresses.

Incremental Migration to a VPC Using ClassicLink

The ClassicLink feature makes it easier to manage an incremental migration to a VPC. ClassicLink allows you to link an EC2-Classic instance to a VPC in your account in the same region, allowing your new VPC resources to communicate with the EC2-Classic instance using private IP addresses. You can then migrate functionality to the VPC one step at a time. This topic provides some basic steps for managing an incremental migration from EC2-Classic to a VPC.

For more information about ClassicLink, see ClassicLink (p. 457).

Topics

- Step 1: Prepare Your Migration Sequence (p. 472)
- Step 2: Create a VPC (p. 472)
- Step 3: Enable Your VPC for ClassicLink (p. 472)

- Step 4: Create an AMI from Your EC2-Classic Instance (p. 472)
- Step 5: Launch an Instance Into Your VPC (p. 473)
- Step 6: Link Your EC2-Classic Instances to Your VPC (p. 474)
- Step 7: Complete the VPC Migration (p. 474)

Step 1: Prepare Your Migration Sequence

To use ClassicLink effectively, you must first identify the components of your application that must be migrated to the VPC, and then confirm the order in which to migrate that functionality.

For example, you have an application that relies on a presentation web server, a backend database server, and authentication logic for transactions. You may decide to start the migration process with the authentication logic, then the database server, and finally, the web server.

Step 2: Create a VPC

To start using a VPC, ensure that you have one in your account. You can create one using one of these methods:

- In your existing AWS account, open the Amazon VPC console and use the VPC wizard to create a new VPC. For more information, see Scenarios for Amazon VPC. Use this option if you want to set up a VPC quickly in your existing EC2-Classic account, using one of the available configuration sets in the wizard. You'll specify this VPC each time you launch an instance.
- In your existing AWS account, open the Amazon VPC console and set up the components of a VPC according to your requirements. For more information, see Your VPC and Subnets. Use this option if you have specific requirements for your VPC, such as a particular number of subnets. You'll specify this VPC each time you launch an instance.

Step 3: Enable Your VPC for ClassicLink

After you've created a VPC, you can enable it for ClassicLink. For more information about ClassicLink, see ClassicLink (p. 457).

To enable a VPC for ClassicLink

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Your VPCs**.
3. Select your VPC, and then select **Enable ClassicLink** from the **Actions** list.
4. In the confirmation dialog box, choose **Yes, Enable**.

Step 4: Create an AMI from Your EC2-Classic Instance

An AMI is a template for launching your instance. You can create your own AMI based on an existing EC2-Classic instance, then use that AMI to launch instances into your VPC.

The method you use to create your AMI depends on the root device type of your instance, and the operating system platform on which your instance runs. To find out the root device type of your instance, go to the **Instances** page, select your instance, and look at the information in the **Root device type** field in the **Description** tab. If the value is `ebs`, then your instance is EBS-backed. If the value is `instance-store`, then your instance is instance store-backed. You can also use the describe-instances AWS CLI command to find out the root device type.

The following table provides options for you to create your AMI based on the root device type of your instance, and the software platform.

Instance Root Device Type	Action
EBS	Create an EBS-backed AMI from your instance. For more information, see Creating an Amazon EBS-Backed Windows AMI (p. 68).
Instance store	Bundle your instance, and then create an instance store-backed AMI from the manifest that's created during bundling. For more information, see Creating an Instance Store-Backed Windows AMI (p. 70).

(Optional) Store Your Data on Amazon EBS Volumes

You can create an Amazon EBS volume and use it to back up and store the data on your instance—like you would use a physical hard drive. Amazon EBS volumes can be attached and detached from any instance in the same Availability Zone. You can detach a volume from your instance in EC2-Classic, and attach it to a new instance that you launch into your VPC in the same Availability Zone.

For more information about Amazon EBS volumes, see the following topics:

- Amazon EBS Volumes (p. 518)
- Creating an Amazon EBS Volume (p. 525)
- Attaching an Amazon EBS Volume to an Instance (p. 529)

To back up the data on your Amazon EBS volume, you can take periodic snapshots of your volume. If you need to, you can restore an Amazon EBS volume from your snapshot. For more information about Amazon EBS snapshots, see the following topics:

- Amazon EBS Snapshots (p. 549)
- Creating an Amazon EBS Snapshot (p. 550)
- Restoring an Amazon EBS Volume from a Snapshot (p. 527)

Step 5: Launch an Instance Into Your VPC

The next step in the migration process is to launch instances into your VPC so that you can start transferring functionality to them. You can use the AMIs that you created in the previous step to launch instances into your VPC. The instances will have the same data and configurations as your existing EC2-Classic instances.

To launch an instance into your VPC using your custom AMI

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the dashboard, choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image** page, select the **My AMIs** category, and select the AMI you created.
4. On the **Choose an Instance Type** page, select the type of instance, and choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, select your VPC from the **Network** list. Select the required subnet from the **Subnet** list. Configure any other details you require, then go through the next pages of the wizard until you reach the **Configure Security Group** page.
6. Select **Select an existing group**, and select the security group you created earlier. Choose **Review and Launch**.
7. Review your instance details, then choose **Launch** to specify a key pair and launch your instance.

For more information about the parameters you can configure in each step of the wizard, see Launching an Instance (p. 207).

After you've launched your instance and it's in the `running` state, you can connect to it and configure it as required.

Step 6: Link Your EC2-Classic Instances to Your VPC

After you've configured your instances and made the functionality of your application available in the VPC, you can use ClassicLink to enable private IP communication between your new VPC instances and your EC2-Classic instances.

To link an instance to a VPC

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select your EC2-Classic instance, then choose **Actions**, **ClassicLink**, and **Link to VPC**.

   Note
   Ensure that your instance is in the `running` state.
4. In the dialog box, select your ClassicLink-enabled VPC (only VPCs that are enabled for ClassicLink are displayed).
5. Select one or more of the VPC's security groups to associate with your instance. When you are done, choose **Link to VPC**.

Step 7: Complete the VPC Migration

Depending on the size of your application and the functionality that must be migrated, repeat steps 4 to 6 until you've moved all the components of your application from EC2-Classic into your VPC.

After you've enabled internal communication between the EC2-Classic and VPC instances, you must update your application to point to your migrated service in your VPC, instead of your service in the EC2-Classic platform. The exact steps for this depend on your application's design. Generally, this includes updating your destination IP addresses to point to the IP addresses of your VPC instances instead of your EC2-Classic instances. You can migrate your Elastic IP addresses that you are currently using in the EC2-Classic platform to the EC2-VPC platform. For more information, see Migrating an Elastic IP Address from EC2-Classic to EC2-VPC (p. 487).

After you've completed this step and you've tested that the application is functioning from your VPC, you can terminate your EC2-Classic instances, and disable ClassicLink for your VPC. You can also clean up any EC2-Classic resources that you may no longer need to avoid incurring charges for them; for example, you can release Elastic IP addresses, and delete the volumes that were associated with your EC2-Classic instances.

Amazon EC2 Instance IP Addressing

We provide your instances with IP addresses and DNS hostnames. These can vary depending on whether you launched the instance in the EC2-Classic platform or in a virtual private cloud (VPC).

For information about the EC2-Classic and EC2-VPC platforms, see Supported Platforms (p. 455). For information about Amazon VPC, see What is Amazon VPC? in the *Amazon VPC User Guide*.

Contents

- Private IP Addresses and Internal DNS Hostnames (p. 475)
- Public IP Addresses and External DNS Hostnames (p. 475)

- Elastic IP Addresses (p. 476)
- Amazon DNS Server (p. 477)
- IP Address Differences Between EC2-Classic and EC2-VPC (p. 477)
- Determining Your Public, Private, and Elastic IP Addresses (p. 477)
- Assigning a Public IP Address (p. 479)
- Multiple Private IP Addresses (p. 480)

Private IP Addresses and Internal DNS Hostnames

A private IP address is an IP address that's not reachable over the Internet. You can use private IP addresses for communication between instances in the same network (EC2-Classic or a VPC). For more information about the standards and specifications of private IP addresses, go to RFC 1918.

When you launch an instance, we allocate a private IP address for the instance using DHCP. Each instance is also given an internal DNS hostname that resolves to the private IP address of the instance; for example, `ip-10-251-50-12.ec2.internal`. You can use the internal DNS hostname for communication between instances in the same network, but we can't resolve the DNS hostname outside the network that the instance is in.

An instance launched in a VPC is given a primary private IP address in the address range of the subnet. For more information, see Subnet Sizing in the *Amazon VPC User Guide*. If you don't specify a primary private IP address when you launch the instance, we select an available IP address in the subnet's range for you. Each instance in a VPC has a default network interface (eth0) that is assigned the primary private IP address. You can also specify additional private IP addresses, known as *secondary private IP addresses*. Unlike primary private IP addresses, secondary private IP addresses can be reassigned from one instance to another. For more information, see Multiple Private IP Addresses (p. 480).

For instances launched in EC2-Classic, we release the private IP address when the instance is stopped or terminated. If you restart your stopped instance, it receives a new private IP address.

For instances launched in a VPC, a private IP address remains associated with the network interface when the instance is stopped and restarted, and is released when the instance is terminated.

If you create a custom firewall configuration in EC2-Classic, you must create a rule in your firewall that allows inbound traffic from port 53 (DNS)—with a destination port from the ephemeral range—from the address of the Amazon DNS server; otherwise, internal DNS resolution from your instances fails. If your firewall doesn't automatically allow DNS query responses, then you'll need to allow traffic from the IP address of the Amazon DNS server. To get the IP address of the Amazon DNS server, use the following command from within your instance:

- **Linux**

```
grep nameserver /etc/resolv.conf
```

- **Windows**

```
ipconfig /all | findstr /c:"DNS Servers"
```

Public IP Addresses and External DNS Hostnames

A public IP address is reachable from the Internet. You can use public IP addresses for communication between your instances and the Internet.

Each instance that receives a public IP address is also given an external DNS hostname; for example, `ec2-203-0-113-25.compute-1.amazonaws.com`. We resolve an external DNS hostname to the public IP address of the instance outside the network of the instance, and to the private IP address of the instance from within the network of the instance. The public IP address is mapped to the primary private IP address through network address translation (NAT). For more information about NAT, go to RFC 1631: The IP Network Address Translator (NAT).

When you launch an instance in EC2-Classic, we automatically assign a public IP address to the instance. You cannot modify this behavior. When you launch an instance into a VPC, your subnet has an attribute that determines whether instances launched into that subnet receive a public IP address. By default, we don't automatically assign a public IP address to an instance that you launch in a nondefault subnet.

You can control whether your instance in a VPC receives a public IP address by doing the following:

- Modifying the public IP addressing attribute of your subnet. For more information, see Modifying Your Subnet's Public IP Addressing Behavior in the *Amazon VPC User Guide*.
- Enabling or disabling the public IP addressing feature during launch, which overrides the subnet's public IP addressing attribute. For more information, see Assigning a Public IP Address (p. 479).

A public IP address is assigned to your instance from Amazon's pool of public IP addresses, and is not associated with your AWS account. When a public IP address is disassociated from your instance, it is released back into the public IP address pool, and you cannot reuse it.

You cannot manually associate or disassociate a public IP address from your instance. Instead, in certain cases, we release the public IP address from your instance, or assign it a new one:

- We release the public IP address for your instance when it's stopped or terminated. Your stopped instance receives a new public IP address when it's restarted.
- We release the public IP address for your instance when you associate an Elastic IP address with your instance, or when you associate an Elastic IP address with the primary network interface (eth0) of your instance in a VPC. When you disassociate the Elastic IP address from your instance, it receives a new public IP address.
- If the public IP address of your instance in a VPC has been released, it will not receive a new one if there is more than one network interface attached to your instance.

If you require a persistent public IP address that can be associated to and from instances as you require, use an Elastic IP address instead. You can allocate your own Elastic IP address, and associate it with your instance. For more information, see Elastic IP Addresses (p. 485).

If your instance is in a VPC and you assign it an Elastic IP address, it receives a DNS hostname if DNS hostnames are enabled. For more information, see Using DNS with Your VPC in the *Amazon VPC User Guide*.

> **Note**
> Instances that access other instances through their public NAT IP address are charged for regional or Internet data transfer, depending on whether the instances are in the same region.

Elastic IP Addresses

An Elastic IP address is a public IP address that you can allocate to your account. You can associate it to and from instances as you require, and it's allocated to your account until you choose to release it. For more information about Elastic IP addresses and how to use them, see Elastic IP Addresses (p. 485).

Amazon DNS Server

Amazon provides a DNS server that resolves DNS hostnames to IP addresses. In EC2-Classic, the Amazon DNS server is located at `172.16.0.23`. In EC2-VPC, the Amazon DNS server is located at the base of your VPC network range plus two. For more information, see Amazon DNS Server in the *Amazon VPC User Guide*.

IP Address Differences Between EC2-Classic and EC2-VPC

The following table summarizes the differences between IP addresses for instances launched in EC2-Classic, instances launched in a default subnet, and instances launched in a nondefault subnet.

Characteristic	EC2-Classic	Default Subnet	Nondefault Subnet
Public IP address (from Amazon's public IP address pool)	Your instance receives a public IP address.	Your instance receives a public IP address by default, unless you specify otherwise during launch, or you modify the subnet's public IP address attribute.	Your instance doesn't receive a public IP address by default, unless you specify otherwise during launch, or you modify the subnet's public IP address attribute.
Private IP address	Your instance receives a private IP address from the EC2-Classic range each time it's started.	Your instance receives a static private IP address from the address range of your default subnet.	Your instance receives a static private IP address from the address range of your subnet.
Multiple IP addresses	We select a single private IP address for your instance; multiple IP addresses are not supported.	You can assign multiple private IP addresses to your instance.	You can assign multiple private IP addresses to your instance.
Network interfaces	IP addresses are associated with the instance; network interfaces aren't supported.	IP addresses are associated with a network interface. Each instance has one or more network interfaces.	IP addresses are associated with a network interface. Each instance has one or more network interfaces.
Elastic IP address	An Elastic IP address is disassociated from your instance when you stop it.	An Elastic IP address remains associated with your instance when you stop it.	An Elastic IP address remains associated with your instance when you stop it.
DNS hostnames	DNS hostnames are enabled by default.	DNS hostnames are enabled by default.	DNS hostnames are disabled by default, except if you've created your VPC using the VPC wizard in the Amazon VPC console.

Determining Your Public, Private, and Elastic IP Addresses

You can use the Amazon EC2 console to determine the private IP addresses, public IP addresses, and Elastic IP addresses of your instances. You can also determine the public and private IP addresses of

your instance from within your instance by using instance metadata. For more information, see Instance Metadata and User Data (p. 160).

To determine your instance's private IP addresses using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select your instance. In the details pane, get the private IP address from the **Private IPs** field, and get the internal DNS hostname from the **Private DNS** field.
4. (VPC only) If you have one or more secondary private IP addresses assigned to network interfaces that are attached to your instance, get those IP addresses from the **Secondary private IPs** field.
5. (VPC only) Alternatively, in the navigation pane, click **Network Interfaces**, and then select the a network interface that's associated with your instance.
6. Get the primary private IP address from the **Primary private IP** field, and the internal DNS hostname from the **Private DNS** field.
7. If you've assigned secondary private IP addresses to the network interface, get those IP addresses from the **Secondary private IPs** field.

To determine your instance's public IP addresses using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select your instance. In the details pane, get the public IP address from the **Public IP** field, and get the external DNS hostname from the **Public DNS** field.
4. If an Elastic IP address has been associated with the instance, get the Elastic IP address from the **Elastic IP** field.

   Note
   If you've associated an Elastic IP address with your instance, the **Public IP** field also displays the Elastic IP address.
5. (VPC only) Alternatively, in the navigation pane, choose **Network Interfaces**, and then select a network interface that's associated with your instance.
6. Get the public IP address from the **Public IPs** field. An asterisk (*) indicates the public IP address or Elastic IP address that's mapped to the primary private IP address.

   Note
   The public IP address is displayed as a property of the network interface in the console, but it's mapped to the primary private IP address through NAT. Therefore, if you inspect the properties of your network interface on your instance, for example, through `ifconfig` (Linux) or `ipconfig` (Windows), the public IP address is not displayed. To determine your instance's public IP address from within the instance, you can use instance metadata.

To determine your instance's IP addresses using instance metadata

1. Connect to your instance.
2. Use the following command to access the private IP address:
   - **Linux**

     ```
     C:\> curl http://169.254.169.254/latest/meta-data/local-ipv4
     ```

   - **Windows**

```
C:\> wget http://169.254.169.254/latest/meta-data/local-ipv4
```

3. Use the following command to access the public IP address:
   - **Linux**

     ```
     C:\> curl http://169.254.169.254/latest/meta-data/public-ipv4
     ```

   - **Windows**

     ```
     C:\> wget http://169.254.169.254/latest/meta-data/public-ipv4
     ```

   Note that if an Elastic IP address is associated with the instance, the value returned is that of the Elastic IP address.

Assigning a Public IP Address

If you launch an instance in EC2-Classic, it is assigned a public IP address by default. You can't modify this behavior.

In a VPC, all subnets have an attribute that determines whether instances launched into that subnet are assigned a public IP address. By default, nondefault subnets have this attribute set to false, and default subnets have this attribute set to true. If you launch an instance into a VPC, a public IP addressing feature is available for you to control whether your instance is assigned a public IP address; you can override the default behavior of the subnet's IP addressing attribute. The public IP address is assigned from Amazon's pool of public IP addresses, and is assigned to the network interface with the device index of eth0. This feature depends on certain conditions at the time you launch your instance.

Important
You can't manually disassociate the public IP address from your instance after launch. Instead, it's automatically released in certain cases, after which you cannot reuse it. For more information, see Public IP Addresses and External DNS Hostnames (p. 475). If you require a persistent public IP address that you can associate or disassociate at will, assign an Elastic IP address to the instance after launch instead. For more information, see Elastic IP Addresses (p. 485).

To access the public IP addressing feature when launching an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Launch Instance**.
3. Select an AMI and an instance type, and then choose **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, select a VPC from the **Network** list. An **Auto-assign Public IP** list is displayed. Choose **Enable** or **Disable** to override the default setting for the subnet.

   Important
   A public IP address can only be assigned to a single, new network interface with the device index of eth0. The **Auto-assign Public IP** list is not available if you're launching with multiple network interfaces, or if you select an existing network interface for eth0.
5. Follow the steps on the next pages of the wizard to complete your instance's setup. For more information about the wizard configuration options, see Launching an Instance (p. 207). On the final **Review Instance Launch** page, review your settings, and then choose **Launch** to choose a key pair and launch your instance.
6. On the **Instances** page, select your new instance and view its public IP address in **Public IP** field in the details pane.

The public IP addressing feature is only available during launch. However, whether you assign a public IP address to your instance during launch or not, you can associate an Elastic IP address with your instance after it's launched. For more information, see Elastic IP Addresses (p. 485). You can also modify your subnet's public IP addressing behavior. For more information, see Modifying Your Subnet's Public IP Addressing Behavior.

API and Command Line Tools for Public IP Addressing

To enable or disable the public IP addressing feature, use one of the methods in the table below. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

Method	Parameter
AWS CLI	Use the `--associate-public-ip-address` or the `--no-associate-public-ip-address` option with the run-instances command.
Amazon EC2 CLI	Use the `--associate-public-ip-address` option with the ec2-run-instances command.
AWS Tools for Windows PowerShell	Use the `-AssociatePublicIp` parameter with the New-EC2Instance command.
Query API	Use the `NetworkInterface.n.AssociatePublicIpAddress` parameter with the RunInstances request.

Multiple Private IP Addresses

In EC2-VPC, you can specify multiple private IP addresses for your instances. The number of network interfaces and private IP addresses that you can specify for an instance depends on the instance type. For more information, see Private IP Addresses Per ENI Per Instance Type (p. 493).

It can be useful to assign multiple private IP addresses to an instance in your VPC to do the following:

- Host multiple websites on a single server by using multiple SSL certificates on a single server and associating each certificate with a specific IP address.
- Operate network appliances, such as firewalls or load balancers, that have multiple private IP addresses for each network interface.
- Redirect internal traffic to a standby instance in case your instance fails, by reassigning the secondary private IP address to the standby instance.

Contents

- How Multiple IP Addresses Work (p. 480)
- Assigning a Secondary Private IP Address (p. 481)
- Configuring the Operating System on Your Instance to Recognize the Secondary Private IP Address (p. 483)
- Associating an Elastic IP Address with the Secondary Private IP Address (p. 483)
- Viewing Your Secondary Private IP Addresses (p. 483)
- Unassigning a Secondary Private IP Address (p. 484)

How Multiple IP Addresses Work

The following list explains how multiple IP addresses work with network interfaces:

- You can assign a secondary private IP address to any network interface. The network interface can be attached to or detached from the instance.
- You must choose a secondary private IP address that's in the CIDR block range of the subnet for the network interface.
- Security groups apply to network interfaces, not to IP addresses. Therefore, IP addresses are subject to the security group of the network interface in which they're specified.
- Secondary private IP addresses can be assigned and unassigned to elastic network interfaces attached to running or stopped instances.
- Secondary private IP addresses that are assigned to a network interface can be reassigned to another one if you explicitly allow it.
- When assigning multiple secondary private IP addresses to a network interface using the command line tools or API, the entire operation fails if one of the secondary private IP addresses can't be assigned.
- Primary private IP addresses, secondary private IP addresses, and any associated Elastic IP addresses remain with the network interface when it is detached from an instance or attached to another instance.
- Although you can't move the primary network interface from an instance, you can reassign the secondary private IP address of the primary network interface to another network interface.
- You can move any additional network interface from one instance to another.

The following list explains how multiple IP addresses work with Elastic IP addresses:

- Each private IP address can be associated with a single Elastic IP address, and vice versa.
- When a secondary private IP address is reassigned to another interface, the secondary private IP address retains its association with an Elastic IP address.
- When a secondary private IP address is unassigned from an interface, an associated Elastic IP address is automatically disassociated from the secondary private IP address.

Assigning a Secondary Private IP Address

You can assign the secondary private IP address to the network interface for an instance as you launch the instance, or after the instance is running.

To assign a secondary private IP address when launching an instance in EC2-VPC

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Launch Instance**.
3. Select an AMI, then choose an instance type and choose **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, choose a VPC from the **Network** list, and a subnet from the **Subnet** list.
5. In the **Network Interfaces** section, do the following, and then choose **Next: Add Storage**:

    a. Choose **Add Device** to add another network interface. The console enables you specify up to two network interfaces when you launch an instance. After you launch the instance, choose **Network Interfaces** in the navigation pane to add additional network interfaces. The total number of network interfaces that you can attach varies by instance type. For more information, see Private IP Addresses Per ENI Per Instance Type (p. 493).

    b. For each network interface, you can specify a primary private IP address, and one or more secondary private IP addresses. For this example, however, accept the IP address that we automatically assign.

    c. Under **Secondary IP addresses**, choose **Add IP**, and then enter a private IP address in the subnet range, or accept the default, `Auto-assign`, to let us select an address.

Important
After you have added a secondary private IP address to a network interface, you must connect to the instance and configure the secondary private IP address on the instance itself. For more information, see Configuring the Operating System on Your Instance to Recognize the Secondary Private IP Address (p. 483).

6. On the next **Add Storage** page, you can specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume), and then choose **Next: Tag Instance**.
7. On the **Tag Instance** page, specify tags for the instance, such as a user-friendly name, and then choose **Next: Configure Security Group**.
8. On the **Configure Security Group** page, select an existing security group or create a new one. Choose **Review and Launch**.
9. On the **Review Instance Launch** page, review your settings, and then choose **Launch** to choose a key pair and launch your instance. If you're new to Amazon EC2 and haven't created any key pairs, the wizard prompts you to create one.

To assign a secondary IP address during launch using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- The `--secondary-private-ip-addresses` option with the run-instances command (AWS CLI)
- The `--secondary-private-ip-address` option with the ec2-run-instances command (Amazon EC2 CLI)

To assign a secondary private IP to an existing instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network Interfaces**, and then select the network interface attached to the instance.
3. Choose **Actions**, and then select **Manage Private IP Addresses**.
4. In the **Manage Private IP Addresses** dialog box, do the following:
   a. Choose **Assign new IP**.
   b. Enter a specific IP address that's within the subnet range for the instance, or leave the field blank and we'll select an IP address for you.
   c. (Optional) Select **Allow reassignment** to allow the secondary private IP address to be reassigned if it is already assigned to another network interface.
   d. Choose **Yes, Update**, and then choose **Close**.

Note that alternatively, you can assign a secondary private IP address to an instance. Choose **Instances** in the navigation pane, select the instance, choose **Actions**, select **Networking**, and then select **Manage Private IP Addresses**. You can configure the same information in the dialog as you did in the steps above.

To assign a secondary private IP to an existing instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- assign-private-ip-addresses (AWS CLI)
- ec2-assign-private-ip-addresses (Amazon EC2 CLI)

- Register-EC2PrivateIpAddress (AWS Tools for Windows PowerShell)

Configuring the Operating System on Your Instance to Recognize the Secondary Private IP Address

After you assign a secondary private IP address to your instance, you need to configure the operating system on your instance to recognize the secondary private IP address.

- If you are using Amazon Linux, the ec2-net-utils package can take care of this step for you. It configures additional network interfaces that you attach while the instance is running, refreshes secondary IP addresses during DHCP lease renewal, and updates the related routing rules. You can immediately refresh the list of interfaces by using the command `sudo service network restart` and then view the up-to-date list using `ip addr li`. If you require manual control over your network configuration, you can remove the ec2-net-utils package.
- If you are using another Linux distribution, see the documentation for your Linux distribution. Search for information about configuring additional network interfaces and secondary IP addresses. If the instance has two or more interfaces on the same subnet, search for information about using routing rules to work around asymmetric routing.
- For information about configuring a Windows instance, see Configuring a Secondary Private IP Address for Your Windows Instance in a VPC (p. 312).

Associating an Elastic IP Address with the Secondary Private IP Address

To associate an Elastic IP address with a secondary private IP address in EC2-VPC

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Actions**, and then select **Associate Address**.
4. In the **Associate Address** dialog box, select the network interface from the **Network Interface** drop-down list, and then select the secondary IP address from the **Private IP address** list.
5. Choose **Associate**.

To associate an Elastic IP address with a secondary private IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- associate-address (AWS CLI)
- ec2-associate-address (Amazon EC2 CLI)
- Register-EC2Address (AWS Tools for Windows PowerShell)

Viewing Your Secondary Private IP Addresses

To view the private IP addresses assigned to a network interface in EC2-VPC

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface whose private IP addresses you want to view.

4. On the **Details** tab in the details pane, check the **Primary private IP** and **Secondary private IPs** fields for the primary private IP address and any secondary private IP addresses assigned to the network interface.

To view the private IP addresses assigned to an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance whose private IP addresses you want to view.
4. On the **Description** tab in the details pane, check the **Private IPs** and **Secondary private IPs** fields for the primary private IP address and any secondary private IP addresses assigned to the instance through its network interface.

Unassigning a Secondary Private IP Address

If you no longer require a secondary private IP address, you can unassign it from the instance or the network interface. When a secondary private IP address is unassigned from an elastic network interface, the Elastic IP address (if it exists) is also disassociated.

To unassign a secondary private IP address from an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select an instance, choose **Actions**, select **Networking**, and then select **Manage Private IP Addresses**.
4. In the **Manage Private IP Addresses** dialog box, beside the secondary private IP address to unassign, choose **Unassign**.
5. Choose **Yes, Update**, and then close the dialog box.

To unassign a secondary private IP address from a network interface

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface, choose **Actions**, and then select **Manage Private IP Addresses**.
4. In the **Manage Private IP Addresses** dialog box, beside the secondary private IP address to unassign, choose **Unassign**.
5. Choose **Yes, Update**, and then choose **Close**.

To unassign a secondary private IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- unassign-private-ip-addresses (AWS CLI)
- ec2-unassign-private-ip-addresses (Amazon EC2 CLI)
- Unregister-EC2PrivateIpAddress (AWS Tools for Windows PowerShell)

Elastic IP Addresses

An *Elastic IP address* is a static IP address designed for dynamic cloud computing. With an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account. Your Elastic IP address is associated with your AWS account, not a particular instance, and it remains associated with your account until you choose to release it explicitly.

If your account supports EC2-Classic, there's one pool of Elastic IP addresses for use with the EC2-Classic platform and another for use with the EC2-VPC platform. You can't associate an Elastic IP address that you allocated for use with a VPC with an instance in EC2-Classic, and vice-versa. However, you can migrate an Elastic IP address you've allocated for use in the EC2-Classic platform to the EC2-VPC platform. For more information about EC2-Classic and EC2-VPC, see Supported Platforms (p. 455).

Topics

- Elastic IP Addresses in EC2-Classic (p. 485)
- Elastic IP Addresses in a VPC (p. 485)
- Elastic IP Address Differences Between EC2-Classic and EC2-VPC (p. 486)
- Migrating an Elastic IP Address from EC2-Classic to EC2-VPC (p. 487)
- Working with Elastic IP Addresses (p. 487)
- Using Reverse DNS for Email Applications (p. 491)
- Elastic IP Address Limit (p. 491)

Elastic IP Addresses in EC2-Classic

By default, we assign each instance in EC2-Classic two IP addresses at launch: a private IP address and a public IP address that is mapped to the private IP address through network address translation (NAT). The public IP address is allocated from the EC2-Classic public IP address pool, and is associated with your instance, not with your AWS account. You cannot reuse a public IP address after it's been disassociated from your instance.

If you use dynamic DNS to map an existing DNS name to a new instance's public IP address, it might take up to 24 hours for the IP address to propagate through the Internet. As a result, new instances might not receive traffic while terminated instances continue to receive requests. To solve this problem, use an Elastic IP address.

When you associate an Elastic IP address with an instance, the instance's current public IP address is released to the EC2-Classic public IP address pool. If you disassociate an Elastic IP address from the instance, the instance is automatically assigned a new public IP address within a few minutes. In addition, stopping the instance also disassociates the Elastic IP address from it.

To ensure efficient use of Elastic IP addresses, we impose a small hourly charge if an Elastic IP address is not associated with a running instance. For more information, see Amazon EC2 Pricing.

Elastic IP Addresses in a VPC

We assign each instance in a default VPC two IP addresses at launch: a private IP address and a public IP address that is mapped to the private IP address through network address translation (NAT). The public IP address is allocated from the EC2-VPC public IP address pool, and is associated with your instance, not with your AWS account. You cannot reuse a public IP address after it's been disassociated from your instance.

We assign each instance in a nondefault VPC only a private IP address, unless you specifically request a public IP address during launch, or you modify the subnet's public IP address attribute. To ensure that an instance in a nondefault VPC that has not been assigned a public IP address can communicate with

the Internet, you must allocate an Elastic IP address for use with a VPC, and then associate that Elastic IP address with the elastic network interface (ENI) attached to the instance.

When you associate an Elastic IP address with an instance in a default VPC, or an instance in which you assigned a public IP to the eth0 network interface during launch, its current public IP address is released to the EC2-VPC public IP address pool. If you disassociate an Elastic IP address from the instance, the instance is automatically assigned a new public IP address within a few minutes. However, if you have attached a second network interface to the instance, the instance is not automatically assigned a new public IP address; you'll have to associate an Elastic IP address with it manually. The Elastic IP address remains associated with the instance when you stop it.

To ensure efficient use of Elastic IP addresss, we impose a small hourly charge if an Elastic IP address is not associated with a running instance, or if it is associated with a stopped instance or an unattached network interface. While your instance is running, you are not charged for one Elastic IP address associated with the instance, but you are charged for any additional Elastic IP addresss associated with the instance. For more information, see Amazon EC2 Pricing.

For information about using an Elastic IP address with an instance in a VPC, see Elastic IP Addresses in the *Amazon VPC User Guide*.

Elastic IP Address Differences Between EC2-Classic and EC2-VPC

The following table lists the differences between Elastic IP addresses on EC2-Classic and EC2-VPC.

Characteristic	EC2-Classic	EC2-VPC
Allocation	When you allocate an Elastic IP address, it's for use in EC2-Classic; however, you can migrate an Elastic IP address to the EC2-VPC platform. For more information, see Migrating an Elastic IP Address from EC2-Classic to EC2-VPC (p. 487).	When you allocate an Elastic IP address, it's for use only in a VPC.
Association	You associate an Elastic IP address with an instance.	An Elastic IP address is a property of an elastic network interface (ENI). You can associate an Elastic IP address with an instance by updating the ENI attached to the instance. For more information, see Elastic Network Interfaces (ENI) (p. 492).
Reassociation	If you try to associate an Elastic IP address that's already associated with another instance, the address is automatically associated with the new instance.	If your account supports EC2-VPC only, and you try to associate an Elastic IP address that's already associated with another instance, the address is automatically associated with the new instance. If you're using a VPC in an EC2-Classic account, and you try to associate an Elastic IP address that's already associated with another instance, it succeeds only if you allowed reassociation.
Instance stop	If you stop an instance, its Elastic IP address is disassociated, and you must re-associate the Elastic IP address when you restart the instance.	If you stop an instance, its Elastic IP address remains associated.

Characteristic	EC2-Classic	EC2-VPC
Multiple IP	Instances support only a single private IP address and a corresponding Elastic IP address.	Instances support multiple IP addresses, and each one can have a corresponding Elastic IP address. For more information, see Multiple Private IP Addresses (p. 480).

Migrating an Elastic IP Address from EC2-Classic to EC2-VPC

If your account supports EC2-Classic, you can migrate Elastic IP addresses that you've allocated for use in the EC2-Classic platform to the EC2-VPC platform, within the same region. This can assist you to migrate your resources from EC2-Classic to a VPC; for example, you can launch new web servers in your VPC, and then use the same Elastic IP addresses that you used for your web servers in EC2-Classic for your new VPC web servers.

After you've migrated an Elastic IP address to EC2-VPC, you cannot use it in the EC2-Classic platform; however, if required, you can restore it to EC2-Classic. After you've restored an Elastic IP address to EC2-Classic, you cannot use it in EC2-VPC until you migrate it again. You can only migrate an Elastic IP address from EC2-Classic to EC2-VPC. You cannot migrate an Elastic IP address that was originally allocated for use in EC2-VPC to EC2-Classic.

> **Note**
> After you've restored an Elastic IP address to EC2-Classic, the address may briefly display in both the EC2-Classic and EC2-VPC platforms.

To migrate an Elastic IP address, it must not be associated with an instance. For more information about disassociating an Elastic IP address from an instance, see Disassociating an Elastic IP Address and Reassociating it with a Different Instance (p. 489).

You can migrate as many EC2-Classic Elastic IP addresses as you can have in your account. However, when you migrate an Elastic IP address to EC2-VPC, it counts against your Elastic IP address limit for EC2-VPC. You cannot migrate an Elastic IP address if it will result in you exceeding your limit. Similarly, when you restore an Elastic IP address to EC2-Classic, it counts against your Elastic IP address limit for EC2-Classic. For more information, see Elastic IP Address Limit (p. 491).

You cannot migrate an Elastic IP address that has been allocated to your account for less than 24 hours.

For more information, see Moving an Elastic IP Address (p. 490).

Working with Elastic IP Addresses

The following sections describe how you can work with Elastic IP addresses.

Topics

- Allocating an Elastic IP Address (p. 488)
- Describing Your Elastic IP Addresses (p. 488)
- Associating an Elastic IP Address with a Running Instance (p. 489)
- Disassociating an Elastic IP Address and Reassociating it with a Different Instance (p. 489)
- Moving an Elastic IP Address (p. 490)
- Releasing an Elastic IP Address (p. 491)

Allocating an Elastic IP Address

You can allocate an Elastic IP address using the Amazon EC2 console or the command line. If your account supports EC2-Classic, you can allocate an address for use in EC2-Classic or in EC2-VPC.

To allocate an Elastic IP address for use in EC2-VPC using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate New Address**.
4. (EC2-Classic accounts) In the **Allocate New Address** dialog box, select **VPC** from **EIP used in**, and then choose **Yes, Allocate**. Close the confirmation dialog box.
5. (VPC-only accounts) Choose **Yes, Allocate**, and close the confirmation dialog box.

To allocate an Elastic IP address for use in EC2-Classic using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate New Address**.
4. Select **EC2**, and then choose **Yes, Allocate**. Close the confirmation dialog box.

To allocate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- allocate-address (AWS CLI)
- ec2-allocate-address (Amazon EC2 CLI)
- New-EC2Address (AWS Tools for Windows PowerShell)

Describing Your Elastic IP Addresses

You can describe an Elastic IP address using the Amazon EC2 or the command line.

To describe your Elastic IP addresses using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Elastic IPs**.
3. Select a filter from the Resource Attribute list to begin searching. You can use multiple filters in a single search.

To describe your Elastic IP addresses using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-addresses (AWS CLI)
- ec2-describe-addresses (Amazon EC2 CLI)
- Get-EC2Address (AWS Tools for Windows PowerShell)

Associating an Elastic IP Address with a Running Instance

You can associate an Elastic IP address to an instance using the Amazon EC2 console or the command line.

To associate an Elastic IP address with an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Elastic IPs**.
3. Select an Elastic IP address, choose **Actions**, and then select **Associate Address**.
4. In the **Associate Address** dialog box, select the instance from **Instance** and then choose **Associate**.

To associate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- associate-address (AWS CLI)
- ec2-associate-address (Amazon EC2 CLI)
- Register-EC2Address (AWS Tools for Windows PowerShell)

Disassociating an Elastic IP Address and Reassociating it with a Different Instance

You can disassociate an Elastic IP address and then reassociate it using the Amazon EC2 console or the command line.

To disassociate and reassociate an Elastic IP address using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address, choose **Actions**, and then select **Disassociate Address**.
4. Choose **Yes, Disassociate** when prompted for confirmation.
5. Select the address that you disassociated in the previous step. For **Actions**, choose **Associate Address**.
6. In the **Associate Address** dialog box, select the new instance from **Instance**, and then choose **Associate**.

To disassociate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- disassociate-address (AWS CLI)
- ec2-disassociate-address (Amazon EC2 CLI)
- Unregister-EC2Address (AWS Tools for Windows PowerShell)

To associate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- associate-address (AWS CLI)
- ec2-associate-address (Amazon EC2 CLI)
- Register-EC2Address (AWS Tools for Windows PowerShell)

Moving an Elastic IP Address

Currently, you can migrate an Elastic IP address to EC2-VPC or restore it to EC2-Classic using the Amazon EC2 Query API, an AWS SDK, or the AWS CLI only.

After you've performed the command to move or restore your Elastic IP address, the process of migrating the Elastic IP address can take a few minutes. Use the describe-moving-addresses command to check whether your Elastic IP address is still moving, or has completed moving.

If the Elastic IP address is in a moving state for longer than 5 minutes, contact http://aws.amazon.com/premiumsupport/.

To move an Elastic IP address using the Amazon EC2 Query API or AWS CLI

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- move-address-to-vpc (AWS CLI)
- MoveAddressToVpc (Amazon EC2 Query API)
- Move-EC2AddressToVpc (AWS Tools for Windows PowerShell)

To restore an Elastic IP address to EC2-Classic using the Amazon EC2 Query API or AWS CLI

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- restore-address-to-classic (AWS CLI)
- RestoreAddressToClassic (Amazon EC2 Query API)
- Restore-EC2AddressToClassic (AWS Tools for Windows PowerShell)

To describe the status of your moving addresses using the Amazon EC2 Query API or AWS CLI

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-moving-addresses (AWS CLI)
- DescribeMovingAddresses (Amazon EC2 Query API)
- Get-EC2Address (AWS Tools for Windows PowerShell)

To retrieve the allocation ID for your migrated Elastic IP address in EC2-VPC

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-addresses (AWS CLI)
- DescribeAddresses (Amazon EC2 Query API)
- Get-EC2Address (AWS Tools for Windows PowerShell)

Releasing an Elastic IP Address

If you no longer need an Elastic IP address, we recommend that you release it (the address must not be associated with an instance). You incur charges for any Elastic IP address that's allocated for use with EC2-Classic but not associated with an instance.

You can release an Elastic IP address using the Amazon EC2 console or the command line.

To release an Elastic IP address using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address, choose **Actions**, and then select **Release Addresses**. Choose **Yes, Release** when prompted.

To release an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- release-address (AWS CLI)
- ec2-release-address (Amazon EC2 CLI)
- Remove-EC2Address (AWS Tools for Windows PowerShell)

Using Reverse DNS for Email Applications

If you intend to send email to third parties from an instance, we suggest you provision one or more Elastic IP addresses and provide them to us. Go to the Create Case page and choose the **EC2 Email** link to get to the *Request to Remove Email Sending Limitations* page. AWS works with ISPs and Internet anti-spam organizations to reduce the chance that your email sent from these addresses will be flagged as spam.

In addition, assigning a static reverse DNS record to your Elastic IP address used to send email can help avoid having email flagged as spam by some anti-spam organizations. You can provide us with a reverse DNS record to associate with your addresses through the aforementioned form. Note that a corresponding forward DNS record (record type A) pointing to your Elastic IP address must exist before we can create your reverse DNS record. After the reverse DNS record is associated with the Elastic IP address, you cannot release the Elastic IP address until the record is removed. Contact AWS account and billing support to remove the reverse DNS record.

> **Note**
> If you have Elastic IP addresses that are associated with reverse DNS records, you cannot migrate them to the EC2-VPC platform yourself, or restore them to the EC2-Classic platform yourself. If you want to migrate these Elastic IP addresses, contact AWS account and billing support.

Elastic IP Address Limit

By default, all AWS accounts are limited to 5 EIPs, because public (IPv4) Internet addresses are a scarce public resource. We strongly encourage you to use an Elastic IP address primarily for the ability to remap the address to another instance in the case of instance failure, and to use DNS hostnames for all other inter-node communication.

If you feel your architecture warrants additional Elastic IP addresses, please complete the Amazon EC2 Elastic IP Address Request Form. We will ask you to describe your use case so that we can understand your need for additional addresses.

Elastic Network Interfaces (ENI)

An elastic network interface (ENI) is a virtual network interface that you can attach to an instance in a VPC. An ENI can include the following attributes:

- a primary private IP address
- one or more secondary private IP addresses
- one Elastic IP address per private IP address
- one public IP address, which can be auto-assigned to the elastic network interface for eth0 when you launch an instance, but only when you create an elastic network interface for eth0 instead of using an existing network interface
- one or more security groups
- a MAC address
- a source/destination check flag
- a description

You can create an elastic network interface, attach it to an instance, detach it from an instance, and attach it to another instance. The attributes of an elastic network interface follow it as it's attached or detached from an instance and reattached to another instance. When you move an elastic network interface from one instance to another, network traffic is redirected to the new instance.

Each instance in a VPC has a default elastic network interface (the primary network interface) that is assigned a private IP address from the IP address range of your VPC. You cannot detach a primary network interface from an instance. You can create and attach additional elastic network interfaces. The maximum number of elastic network interfaces that you can use varies by instance type. For more information, see Private IP Addresses Per ENI Per Instance Type (p. 493).

Attaching multiple elastic network interfaces to an instance is useful when you want to:

- Create a management network.
- Use network and security appliances in your VPC.
- Create dual-homed instances with workloads/roles on distinct subnets.
- Create a low-budget, high-availability solution.

Contents

- Private IP Addresses Per ENI Per Instance Type (p. 493)
- Creating a Management Network (p. 495)
- Use Network and Security Appliances in Your VPC (p. 495)
- Creating Dual-homed Instances with Workloads/Roles on Distinct Subnets (p. 495)
- Create a Low Budget High Availability Solution (p. 496)
- Monitoring IP Traffic on Your Network Interface (p. 496)
- Best Practices for Configuring Elastic Network Interfaces (p. 496)
- Creating an Elastic Network Interface (p. 496)
- Deleting an Elastic Network Interface (p. 497)
- Viewing Details about an Elastic Network Interface (p. 497)
- Attaching an Elastic Network Interface When Launching an Instance (p. 498)
- Attaching an Elastic Network Interface to a Stopped or Running Instance (p. 499)
- Detaching an Elastic Network Interface from an Instance (p. 500)

- Changing the Security Group of an Elastic Network Interface (p. 500)
- Changing the Source/Destination Checking of an Elastic Network Interface (p. 501)
- Associating an Elastic IP Address with an Elastic Network Interface (p. 502)
- Disassociating an Elastic IP Address from an Elastic Network Interface (p. 502)
- Changing Termination Behavior for an Elastic Network Interface (p. 503)
- Adding or Editing a Description for an Elastic Network Interface (p. 503)
- Adding or Editing Tags for an Elastic Network Interface (p. 504)

Private IP Addresses Per ENI Per Instance Type

The following table lists the maximum number of elastic network interfaces (ENI) per instance type, and the maximum number of private IP addresses per ENI. ENIs and multiple private IP addresses are only available for instances running in a VPC. For more information, see Multiple Private IP Addresses (p. 480).

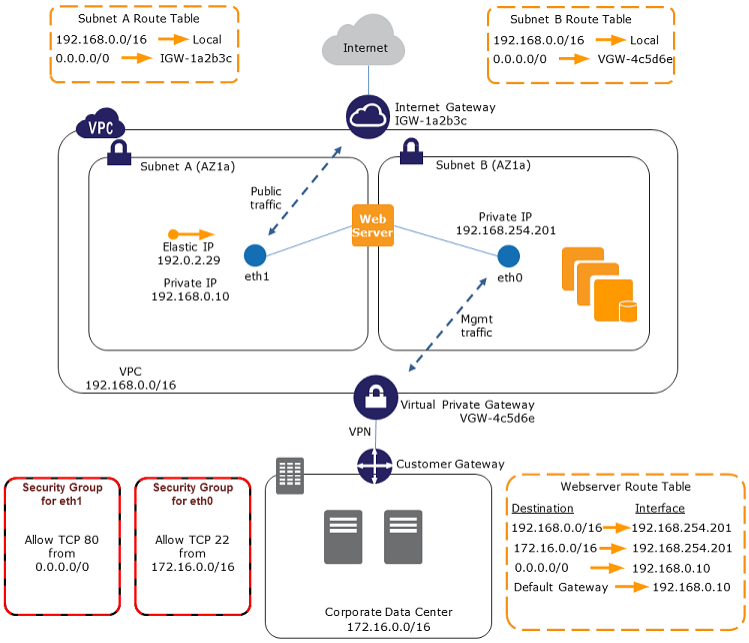
Instance Type	Maximum Elastic Network Interfaces	IP Addresses per Interface
c1.medium	2	6
c1.xlarge	4	15
c3.large	3	10
c3.xlarge	4	15
c3.2xlarge	4	15
c3.4xlarge	8	30
c3.8xlarge	8	30
c4.large	3	10
c4.xlarge	4	15
c4.2xlarge	4	15
c4.4xlarge	8	30
c4.8xlarge	8	30
cc2.8xlarge	8	30
cg1.4xlarge	8	30
cr1.8xlarge	8	30
d2.xlarge	4	15
d2.2xlarge	4	15
d2.4xlarge	8	30
d2.8xlarge	8	30
g2.2xlarge	4	15
g2.8xlarge	8	30
hi1.4xlarge	8	30

Instance Type	Maximum Elastic Network Interfaces	IP Addresses per Interface
hs1.8xlarge	8	30
i2.xlarge	4	15
i2.2xlarge	4	15
i2.4xlarge	8	30
i2.8xlarge	8	30
m1.small	2	4
m1.medium	2	6
m1.large	3	10
m1.xlarge	4	15
m2.xlarge	4	15
m2.2xlarge	4	30
m2.4xlarge	8	30
m3.medium	2	6
m3.large	3	10
m3.xlarge	4	15
m3.2xlarge	4	30
m4.large	2	10
m4.xlarge	4	15
m4.2xlarge	4	15
m4.4xlarge	8	30
m4.10xlarge	8	30
r3.large	3	10
r3.xlarge	4	15
r3.2xlarge	4	15
r3.4xlarge	8	30
r3.8xlarge	8	30
t1.micro	2	2
t2.micro	2	2
t2.small	2	4
t2.medium	3	6
t2.large	3	12

Creating a Management Network

You can create a management network using elastic network interfaces. In this scenario, the secondary elastic network interface on the instance handles public-facing traffic and the primary elastic network interface handles back-end management traffic and is connected to a separate subnet in your VPC that has more restrictive access controls. The public facing interface, which may or may not be behind a load balancer, has an associated security group that allows access to the server from the Internet (for example, allow TCP port 80 and 443 from `0.0.0.0/0`, or from the load balancer) while the private facing interface has an associated security group allowing RDP access only from an allowed range of IP addresses either within the VPC or from the Internet, a private subnet within the VPC or a virtual private gateway.

To ensure failover capabilities, consider using a secondary private IP for incoming traffic on an elastic network interface. In the event of an instance failure, you can move the interface and/or secondary private IP address to a standby instance.



Use Network and Security Appliances in Your VPC

Some network and security appliances, such as load balancers, network address translation (NAT) servers, and proxy servers prefer to be configured with multiple elastic network interfaces. You can create and attach secondary elastic network interfaces to instances in a VPC that are running these types of applications and configure the additional interfaces with their own public and private IP addresses, security groups, and source/destination checking.

Creating Dual-homed Instances with Workloads/Roles on Distinct Subnets

You can place an elastic network interface on each of your web servers that connects to a mid-tier network where an application server resides. The application server can also be dual-homed to a back-end network (subnet) where the database server resides. Instead of routing network packets through the dual-homed instances, each dual-homed instance receives and processes requests on the front end, initiates a connection to the back end, and then sends requests to the servers on the back-end network.

Create a Low Budget High Availability Solution

If one of your instances serving a particular function fails, its elastic network interface can be attached to a replacement or hot standby instance pre-configured for the same role in order to rapidly recover the service. For example, you can use an ENI as your primary or secondary network interface to a critical service such as a database instance or a NAT instance. If the instance fails, you (or more likely, the code running on your behalf) can attach the ENI to a hot standby instance. Because the interface maintains its private IP addresses, Elastic IP addresses, and MAC address, network traffic will begin flowing to the standby instance as soon as you attach the ENI to the replacement instance. Users will experience a brief loss of connectivity between the time the instance fails and the time that the ENI is attached to the standby instance, but no changes to the VPC route table or your DNS server are required.

Monitoring IP Traffic on Your Network Interface

You can enable a VPC flow log on your elastic network interface to capture information about the IP traffic going to and from the interface. After you've created a flow log, you can view and retrieve its data in Amazon CloudWatch Logs.

For more information about flow logs, see VPC Flow Logs in the *Amazon VPC User Guide*.

Best Practices for Configuring Elastic Network Interfaces

- You can attach an elastic network interface to an instance when it's running (hot attach), when it's stopped (warm attach), or when the instance is being launched (cold attach).
- You can detach secondary (eth*N*) elastic network interfaces when the instance is running or stopped. However, you can't detach the primary (eth0) interface.
- You can attach an elastic network interface in one subnet to an instance in another subnet in the same VPC; however, both the elastic network interface and the instance must reside in the same Availability Zone.
- When launching an instance from the CLI or API, you can specify the elastic network interfaces to attach to the instance for both the primary (eth0) and additional elastic network interfaces.
- Launching an Amazon Linux or Microsoft Windows Server instance with multiple network interfaces automatically configures interfaces, private IP addresses, and route tables on the operating system of the instance.
- A warm or hot attach of an additional elastic network interface may require you to manually bring up the second interface, configure the private IP address, and modify the route table accordingly. Instances running Amazon Linux or Microsoft Windows Server automatically recognize the warm or hot attach and configure themselves.
- Attaching another elastic network interface to an instance is not a method to increase or double the network bandwidth to or from the dual-homed instance.

Creating an Elastic Network Interface

You can create an elastic network interface using the Amazon EC2 console or the command line.

To create an elastic network interface using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network Interfaces**.
3. Choose **Create Network Interface**.

4. In the **Create Network Interface** dialog box, provide the following information for the elastic network interface, and then choose **Yes, Create**.
   a. In **Description**, enter a descriptive name.
   b. In **Subnet**, select the subnet. Note that you can't move the elastic network interface to another subnet after it's created, and you can only attach the interface to instances in the same Availability Zone.
   c. In **Private IP**, enter the primary private IP address. If you don't specify an IP address, we'll select an available private IP address from within the selected subnet.
   d. In **Security groups**, select one or more security groups.

To create an elastic network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- create-network-interface (AWS CLI)
- ec2-create-network-interface (Amazon EC2 CLI)
- New-EC2NetworkInterface (AWS Tools for Windows PowerShell)

Deleting an Elastic Network Interface

You must first detach an elastic network interface from an instance before you can delete it. Deleting an elastic network interface releases all attributes associated with the interface and releases any private IP addresses or Elastic IP addresses to be used by another instance.

You can delete an elastic network interface using the Amazon EC2 console or the command line.

To delete an elastic network interface using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network Interfaces**.
3. Select an elastic network interface, and then choose **Delete**.
4. In the **Delete Network Interface** dialog box, choose **Yes, Delete**.

To delete an elastic network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- delete-network-interface (AWS CLI)
- ec2-delete-network-interface (Amazon EC2 CLI)
- Remove-EC2NetworkInterface (AWS Tools for Windows PowerShell)

Viewing Details about an Elastic Network Interface

You can describe an elastic network interface using the Amazon EC2 or the command line.

To describe an elastic network interface using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface.
4. View the details on the **Details** tab.

To describe an elastic network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-network-interfaces (AWS CLI)
- ec2-describe-network-interfaces (Amazon EC2 CLI)
- Get-EC2NetworkInterface (AWS Tools for Windows PowerShell)

To describe an elastic network interface attribute using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-network-interface-attribute (AWS CLI)
- ec2-describe-network-interface-attribute (Amazon EC2 CLI)
- Get-EC2NetworkInterfaceAttribute (AWS Tools for Windows PowerShell)

Attaching an Elastic Network Interface When Launching an Instance

You can attach an additional elastic network interface to an instance when you launch it into a VPC. You can do this using the Amazon EC2 or the command line.

> **Note**
> If an error occurs when attaching an elastic network interface to your instance, this causes the instance launch to fail.

To attach an elastic network interface when launching an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Launch Instance**.
3. Choose an AMI, then choose an instance type and choose **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, select a VPC from the **Network** list, and a subnet from the **Subnet** list.

   To assign a public IP address to your instance, select **Enable** from the **Auto-assign Public IP** list (if you selected a default subnet, you can leave the **Use subnet setting** option). Note that you can't assign a public IP address to your instance if you specify an existing elastic network interface for the primary elastic network interface (eth0) or multiple elastic network interfaces in the next step.
5. In the **Network Interfaces** section, the console enables you specify up to 2 elastic network interfaces (new, existing, or a combination) when you launch an instance. You can also enter a primary IP address and one or more secondary IP addresses for any new interface. When you've finished, choose **Next: Add Storage**.

Note that you can add additional network interfaces to the instance after you launch it. The total number of network interfaces that you can attach varies by instance type. For more information, see Private IP Addresses Per ENI Per Instance Type (p. 493).

6. On the next **Add Storage** page, you can specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume), and then choose **Next: Tag Instance**.
7. On the **Tag Instance** page, specify tags for the instance, such as a user-friendly name, and then click **Next: Configure Security Group**.
8. On the **Configure Security Group** page, select an existing security group or create a new one. Choose **Review and Launch**.
9. On the **Review Instance Launch** page, details about the primary and additional network interface are displayed. Review the settings, and then choose **Launch** to choose a key pair and launch your instance. If you're new to Amazon EC2 and haven't created any key pairs, the wizard prompts you to create one.

To attach an elastic network interface when launching an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- run-instances (AWS CLI)
- ec2-run-instances (Amazon EC2 CLI)
- New-EC2Instance (AWS Tools for Windows PowerShell)

Attaching an Elastic Network Interface to a Stopped or Running Instance

You can attach an elastic network interface to any of your stopped or running instances in your VPC using either the **Instances** or **Network Interfaces** page of the Amazon EC2 console, or using a command line interface.

Note
If the public IP address on your instance is released, it will not receive a new one if there is more than one elastic network interface attached to the instance. For more information about the behavior of public IP addresses, see Public IP Addresses and External DNS Hostnames (p. 475).

To attach an elastic network interface to an instance using the Instances page

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Choose **Actions**, select **Networking**, and then select **Attach Network Interface**.
4. In the **Attach Network Interface** dialog box, select the elastic network interface, and then choose **Attach**.

To attach an elastic network interface to an instance using the Network Interfaces page

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface, and then choose **Attach**.
4. In the **Attach Network Interface** dialog box, select the instance, and then choose **Attach**.

To attach an elastic network interface to an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- attach-network-interface (AWS CLI)
- ec2-attach-network-interface (Amazon EC2 CLI)
- Add-EC2NetworkInterface (AWS Tools for Windows PowerShell)

Detaching an Elastic Network Interface from an Instance

You can detach a secondary elastic network interface at any time, using either the **Instances** or **Network Interfaces** page of the Amazon EC2 console, or using a command line interface.

To detach an elastic network interface from an instance using the Instances page

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Choose **Actions**, select **Networking**, and then select **Detach Network Interface**.
4. In the **Detach Network Interface** dialog box, select the elastic network interface, and then choose **Detach**.

To detach an elastic network interface from an instance using the Network Interfaces page

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface, and then choose **Detach**.
4. In the **Detach Network Interface** dialog box, choose **Yes, Detach**. If the elastic network interface fails to detach from the instance, select **Force detachment**, and then try again.

To detach an elastic network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- detach-network-interface (AWS CLI)
- ec2-detach-network-interface (Amazon EC2 CLI)
- Dismount-EC2NetworkInterface (AWS Tools for Windows PowerShell)

Changing the Security Group of an Elastic Network Interface

You can change the security groups that are associated with an elastic network interface. When you create the security group, be sure to specify the same VPC as the subnet for the interface.

You can change the security group for your elastic network interfaces using the Amazon EC2 or the command line.

Note
To change security group membership for interfaces owned by other Amazon Web Services, such as Elastic Load Balancing, use the console or command line interface for that service.

To change the security group of an elastic network interface using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface, choose **Actions**, and then select **Change Security Groups**.
4. In the **Change Security Groups** dialog box, select the security groups to use, and then choose **Save**.

To change the security group of an elastic network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- modify-network-interface-attribute (AWS CLI)
- ec2-modify-network-interface-attribute (Amazon EC2 CLI)
- Edit-EC2NetworkInterfaceAttribute (AWS Tools for Windows PowerShell)

Changing the Source/Destination Checking of an Elastic Network Interface

The Source/Destination Check attribute controls whether source/destination checking is enabled on the instance. Disabling this attribute enables an instance to handle network traffic that isn't specifically destined for the instance. For example, instances running services such as network address translation, routing, or a firewall should set this value to `disabled`. The default value is `enabled`.

You can change source/destination checking using the Amazon EC2 or the command line.

To change source/destination checking for an elastic network interface using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface, choose **Actions**, and then select **Change Source/Dest Check**.
4. In the dialog box, select **Enabled** (if enabling), or **Disabled** (if disabling), and then choose **Save**.

To change source/destination checking for an elastic network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- modify-network-interface-attribute (AWS CLI)
- ec2-modify-network-interface-attribute (Amazon EC2 CLI)
- Edit-EC2NetworkInterfaceAttribute (AWS Tools for Windows PowerShell)

Associating an Elastic IP Address with an Elastic Network Interface

If you have an Elastic IP address, you can associate it with one of the private IP addresses for the elastic network interface. You can associate one Elastic IP address with each private IP address.

You can associate an Elastic IP address using the Amazon EC2 or the command line.

To associate an Elastic IP address using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface, choose **Actions**, and then select **Associate Address**.
4. In the **Associate Elastic IP Address** dialog box, select the Elastic IP address from the **Address** list.
5. In **Associate to private IP address**, select the private IP address to associate with the Elastic IP address.
6. Select **Allow reassociation** to allow the Elastic IP address to be associated with the specified network interface if it's currently associated with another instance or network interface, and then choose **Associate Address**.

To associate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- associate-address (AWS CLI)
- ec2-associate-address (Amazon EC2 CLI)
- Register-EC2Address (AWS Tools for Windows PowerShell)

Disassociating an Elastic IP Address from an Elastic Network Interface

If the elastic network interface has an Elastic IP address associated with it, you can disassociate the address, and then either associate it with another elastic network interface or release it back to the address pool. Note that this is the only way to associate an Elastic IP address with an instance in a different subnet or VPC using an elastic network interface, as elastic network interfaces are specific to a particular subnet.

You can disassociate an Elastic IP address using the Amazon EC2 or the command line.

To disassociate an Elastic IP address using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface, choose **Actions**, and then select **Disassociate Address**.
4. In the **Disassociate IP Address** dialog box, choose **Yes, Disassociate**.

To disassociate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- disassociate-address (AWS CLI)
- ec2-disassociate-address (Amazon EC2 CLI)
- Unregister-EC2Address (AWS Tools for Windows PowerShell)

Changing Termination Behavior for an Elastic Network Interface

You can set the termination behavior for an elastic network interface attached to an instance so that it is automatically deleted when you delete the instance to which it's attached.

> **Note**
> By default, elastic network interfaces that are automatically created and attached to instances using the console are set to terminate when the instance terminates. However, network interfaces created using the command line interface aren't set to terminate when the instance terminates.

You can change the terminating behavior for an elastic network interface using the Amazon EC2 or the command line.

To change the termination behavior for an elastic network interface using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface, choose **Actions**, and then select **Change Termination Behavior**.
4. In the **Change Termination Behavior** dialog box, select the **Delete on termination** check box if you want the elastic network interface to be deleted when you terminate an instance.

To change the termination behavior for an elastic network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- modify-network-interface-attribute (AWS CLI)
- ec2-modify-network-interface-attribute (Amazon EC2 CLI)
- Edit-EC2NetworkInterfaceAttribute (AWS Tools for Windows PowerShell)

Adding or Editing a Description for an Elastic Network Interface

You can change the description for an elastic network interface using the Amazon EC2 or the command line.

To change the description for an elastic network interface using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface, choose **Actions**, and then select **Change Description**.
4. In the **Change Description** dialog box, enter a description for the elastic network interface, and then choose **Save**.

To change the description for an elastic network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- modify-network-interface-attribute (AWS CLI)
- ec2-modify-network-interface-attribute (Amazon EC2 CLI)
- Edit-EC2NetworkInterfaceAttribute (AWS Tools for Windows PowerShell)

Adding or Editing Tags for an Elastic Network Interface

Tags are metadata that you can add to an elastic network interface. Tags are private and are only visible to your account. Each tag consists of a key and an optional value. For more information about tags, see Tagging Your Amazon EC2 Resources (p. 609).

You can tag a resource using the Amazon EC2 or the command line.

To add or edit tags for an elastic network interface using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the elastic network interface.
4. In the details pane, choose the **Tags** tab, and then choose **Add/Edit Tags**.
5. In the **Add/Edit Tags** dialog, choose **Create Tag** for each tag you want to create, and enter a key and optional value. When you're done, choose **Save**.

To add or edit tags for an elastic network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- create-tags (AWS CLI)
- ec2-create-tags (Amazon EC2 CLI)
- New-EC2Tag (AWS Tools for Windows PowerShell)

Placement Groups

A *placement group* is a logical grouping of instances within a single Availability Zone. Using placement groups enables applications to participate in a low-latency, 10 Gbps network. Placement groups are recommended for applications that benefit from low network latency, high network throughput, or both. To provide the lowest latency, and the highest packet-per-second network performance for your placement group, choose an instance type that supports enhanced networking. For more information, see Enhanced Networking (p. 510).

First, you create a placement group and then you launch multiple instances into the placement group. We recommend that you launch the number of instances that you need in the placement group in a single launch request and that you use the same instance type for all instances in the placement group. If you try to add more instances to the placement group later, or if you try to launch more than one instance type in the placement group, you increase your chances of getting an insufficient capacity error.

If you stop an instance in a placement group and then start it again, it still runs in the placement group. However, the start fails if there isn't enough capacity for the instance.

If you receive a capacity error when launching an instance in a placement group, stop and restart the instances in the placement group, and then try the launch again.

Contents

- Placement Group Limitations (p. 505)
- Launching Instances into a Placement Group (p. 505)
- Deleting a Placement Group (p. 507)

Placement Group Limitations

Placement groups have the following limitations:

- A placement group can't span multiple Availability Zones.
- The name you specify for a placement group a name must be unique within your AWS account.
- The following are the only instance types that you can use when you launch an instance into a placement group:
  - General purpose: `m4.large` | `m4.xlarge` | `m4.2xlarge` | `m4.4xlarge` | `m4.10xlarge`
  - Compute optimized: `c4.large` | `c4.xlarge` | `c4.2xlarge` | `c4.4xlarge` | `c4.8xlarge` | `c3.large` | `c3.xlarge` | `c3.2xlarge` | `c3.4xlarge` | `c3.8xlarge` | `cc2.8xlarge`
  - Memory optimized: `cr1.8xlarge` | `r3.large` | `r3.xlarge` | `r3.2xlarge` | `r3.4xlarge` | `r3.8xlarge`
  - Storage optimized: `d2.xlarge` | `d2.2xlarge` | `d2.4xlarge` | `d2.8xlarge` | `hi1.4xlarge` | `hs1.8xlarge` | `i2.xlarge` | `i2.2xlarge` | `i2.4xlarge` | `i2.8xlarge`
  - GPU: `cg1.4xlarge` | `g2.2xlarge` | `g2.8xlarge`
- Not all of the instance types that can be launched into a placement group can take full advantage of the 10 gigabit network speeds provided. Instance types that support 10 gigabit network speeds are listed in the Amazon EC2 Instance Types Matrix.
- Although launching multiple instance types into a placement group is possible, this reduces the likelihood that the required capacity will be available for your launch to succeed. We recommend using the same instance type for all instances in a placement group.
- You can't merge placement groups. Instead, you must terminate the instances in one placement group, and then relaunch those instances into the other placement group.
- A placement group can span peered VPCs; however, you will not get full-bisection bandwidth between instances in peered VPCs. For more information about VPC peering connections, see VPC Peering in the *Amazon VPC User Guide*.
- You can't move an existing instance into a placement group. You can create an AMI from your existing instance, and then launch a new instance from the AMI into a placement group.

Launching Instances into a Placement Group

We suggest that you create an AMI specifically for the instances that you'll launch into a placement group.

To launch instances into a placement group using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Create an AMI for your instances.

a. From the Amazon EC2 dashboard, click **Launch Instance**. After you complete the wizard, click **Launch**.
b. Connect to your instance. (For more information, see Connecting to Your Windows Instance Using RDP (p. 216).)
c. Install software and applications on the instance, copy data, or attach additional Amazon EBS volumes.
d. (Optional) If your instance type supports enhanced networking, ensure that this feature is enabled by following the procedures in Enabling Enhanced Networking on Windows Instances in a VPC (p. 510).
e. In the navigation pane, click **Instances**, select your instance, click **Actions**, select **Image**, and then click **Create Image**. Provide the information requested by the **Create Image** dialog box, and then click **Create Image**.
f. (Optional) You can terminate this instance if you have no further use for it.

3. Create a placement group.

a. In the navigation pane, click **Placement Groups**.
b. Click **Create Placement Group**.
c. In the **Create Placement Group** dialog box, provide a name for the placement group that is unique in the AWS account you're using, and then click **Create**.

When the status of the placement group is `available`, you can launch instances into the placement group.

4. Launch instances into your placement group.

a. In the navigation pane, click **Instances**.
b. Click **Launch Instance**. Complete the wizard as directed, taking care to do the following:

- On the **Choose an Amazon Machine Image (AMI)** page, select the **My AMIs** tab, and then select the AMI that you created.
- On the **Choose an Instance Type** page, select an instance type that can be launched into a placement group.
- On the **Configure Instance Details** page, enter the total number of instances that you'll need in this placement group, as you might not be able to add instances to the placement group later on.
- On the **Configure Instance Details** page, select the placement group that you created from **Placement group**. If you do not see the **Placement group** list on this page, verify that you have selected an instance type that can be launched into a placement group, as this option is not available otherwise.

To launch instances into a placement group using the command line

1. Create an AMI for your instances using one of the following commands:

- create-image (AWS CLI)
- ec2-create-image (Amazon EC2 CLI)
- New-EC2Image (AWS Tools for Windows PowerShell)

2. Create a placement group using one of the following commands:

   - create-placement-group (AWS CLI)
   - ec2-create-placement-group (Amazon EC2 CLI)
   - New-EC2PlacementGroup (AWS Tools for Windows PowerShell)

3. Launch instances into your placement group using one of the following options:

   - `--placement` with run-instances (AWS CLI)
   - `--placement-group` with ec2-run-instances (Amazon EC2 CLI)
   - `-PlacementGroup` with New-EC2Instance (AWS Tools for Windows PowerShell)

Deleting a Placement Group

You can delete a placement group if you need to replace it or no longer need a placement group. Before you can delete your placement group, you must terminate all instances that you launched into the placement group.

To delete a placement group using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Instances**.
3. Select and terminate all instances in the placement group. (You can verify that the instance is in a placement group before you terminate it by checking the value of **Placement Group** in the details pane.)
4. In the navigation pane, click **Placement Groups**.
5. Select the placement group, and then click **Delete Placement Group**.
6. When prompted for confirmation, click **Yes, Delete**.

To delete a placement group using the command line

You can use one of the following sets of commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- terminate-instances and delete-placement-group (AWS CLI)
- ec2-terminate-instances and ec2-delete-placement-group (Amazon EC2 CLI)
- Stop-EC2Instance and Remove-EC2PlacementGroup(AWS Tools for Windows PowerShell)

Network Maximum Transmission Unit (MTU) for Your EC2 Instance

The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection. The larger the MTU of a connection, the more data that can be passed in a single packet. Ethernet packets consist of the frame, or the actual data you are sending, and the network overhead information that surrounds it.

Ethernet frames can come in different formats, and the most common format is the standard Ethernet v2 frame format. It supports 1500 MTU, which is the largest Ethernet packet size supported over most of

the Internet. The maximum supported MTU for an instance depends on its instance type. All Amazon EC2 instance types support 1500 MTU, and many current instance sizes support 9001 MTU, or jumbo frames.

Contents

- Jumbo Frames (9001 MTU) (p. 508)
- Path MTU Discovery (p. 508)
- Check the Path MTU Between Two Hosts (p. 509)
- Check and Set the MTU on your Amazon EC2 Instance (p. 509)
- Troubleshooting (p. 510)

Jumbo Frames (9001 MTU)

Jumbo frames allow more than 1500 bytes of data by increasing the payload size per packet, and thus increasing the percentage of the packet that is not packet overhead. Fewer packets are needed to send the same amount of usable data. However, outside of a given AWS region (EC2-Classic) or a single VPC, you will experience a maximum path of 1500 MTU. VPN connections, VPC peering connections, and traffic sent over an Internet gateway are limited to 1500 MTU. If packets are over 1500 bytes, they are fragmented, or they are dropped if the `Don't Fragment` flag is set in the IP header.

Jumbo frames should be used with caution for Internet-bound traffic or any traffic that leaves a VPC. Packets are fragmented by intermediate systems, which slows down this traffic. To use jumbo frames inside a VPC and not slow traffic that's bound for outside the VPC, you can configure the MTU size by route, or use multiple elastic network interfaces with different MTU sizes and different routes.

For instances that are collocated inside a placement group, jumbo frames help to achieve the maximum network throughput possible, and they are recommended in this case. For more information, see Placement Groups (p. 504).

The following instances support jumbo frames:

- Compute optimized: C3, C4, CC2
- General purpose: M3, M4, T2
- GPU: CG1, G2
- Memory optimized: CR1, R3
- Storage optimized: D2, HI1, HS1, I2

Path MTU Discovery

Path MTU Discovery is used to determine the path MTU between two devices. The path MTU is the maximum packet size that's supported on the path between the originating host and the receiving host. If a host sends a packet that's larger than the MTU of the receiving host or that's larger than the MTU of a device along the path, the receiving host or device returns the following ICMP message: `Destination Unreachable: Fragmentation Needed and Don't Fragment was Set` (Type 3, Code 4). This instructs the original host to adjust the MTU until the packet can be transmitted.

By default, security groups do not allow any inbound ICMP traffic. To ensure that your instance can receive this message and the packet does not get dropped, you must add a **Custom ICMP Rule** with the **Destination Unreachable** protocol to the inbound security group rules for your instance. For more information, see the Adding Rules to a Security Group (p. 403) and API and Command Overview (p. 405) sections in the Amazon EC2 Security Groups topic.

Important
Modifying your instance's security group to allow path MTU discovery does not guarantee that jumbo frames will not be dropped by some routers. An Internet gateway in your VPC will forward packets up to 1500 bytes only. 1500 MTU packets are recommended for Internet traffic.

Check the Path MTU Between Two Hosts

You can check the path MTU between two hosts using the **mturoute.exe** command, which you can download and install from http://www.elifulkerson.com/projects/mturoute.php.

To check path MTU with mturoute.exe

1. Download **mturoute.exe** from http://www.elifulkerson.com/projects/mturoute.php.
2. Open a command prompt window and change to the directory where you downloaded **mturoute.exe**.
3. Use the following command to check the path MTU between your Amazon EC2 instance and another host. You can use a DNS name or an IP address as the destination; this example checks the path MTU between an EC2 instance and `www.elifulkerson.com`.

```
PS C:\Users\Administrator\Downloads> .\mturoute.exe www.elifulkerson.com
* ICMP Fragmentation is not permitted. *
* Speed optimization is enabled. *
* Maximum payload is 10000 bytes. *
+ ICMP payload of 1472 bytes succeeded.
- ICMP payload of 1473 bytes is too big.
Path MTU: 1500 bytes.
```

In this example, the path MTU is 1500.

Check and Set the MTU on your Amazon EC2 Instance

Some AMIs are configured to use jumbo frames on instance that support them, and others are configured to use standard frame sizes. You may want to use jumbo frames for network traffic within your VPC or you may want to use standard frames for Internet traffic. Whatever your use case, we recommend verifying that your instance will behave the way you expect it to. You can use the procedures in this section to check your network interface's MTU setting and modify it if needed.

To check the MTU setting on a Windows instance

- If your instance uses a Windows operating system, you can review the MTU value with the **netsh** command. Run the following command to determine the current MTU value:

```
PS C:\Users\Administrator> netsh interface ipv4 show subinterface

   MTU  MediaSenseState   Bytes In  Bytes Out  Interface
------  ---------------  ---------  ---------  -------------
  9001                1     317337     692805  Ethernet
```

In the resulting output, look for the entry titled "Ethernet," "Ethernet 2," or "Local Area Connection."

In the above example, the *9001* in the **MTU** column indicates that this instance uses jumbo frames.

To set the MTU value on a Windows instance

1. If your instance uses a Windows operating system, you can set the MTU value with the **netsh** command. Run the following command to set the desired MTU value. This procedure sets the MTU to 1500, but it is the same for 9001.

   Note
   These steps vary based on the network drivers your Windows instance uses; make sure to execute the correct command for your driver version. For more information, see Paravirtual Drivers (p. 258).

   - For Windows instances that use AWS PV drivers or the Intel network driver for enhanced networking, execute the following command to set the MTU to 1500.

     ```
     PS C:\Users\Administrator> netsh interface ipv4 set subinterface "Ethernet" mtu=1500 store=persistent
     Ok.
     ```

     Note
     If you receive an `Element not found` error, replace *`Ethernet`* with the `Interface` column output from the To check the MTU setting on a Windows instance (p. 509) procedure that matches your interface.

   - For Windows instances that use Citrix PV drivers, first ensure that your PV drivers are up to date by following the procedures in Upgrading PV Drivers on Your Windows AMI (p. 265). Then, execute the following command to set the MTU to 1500. Citrix PV drivers interpret MTU to mean max frame size, so you must subtract 18 from your `mtu` setting to set the correct value. For example, to set 1500 MTU, use 1482 in the command below, and to set 9001 MTU, use 8983 instead.

     ```
     PS C:\Users\Administrator> netsh interface ipv4 set subinterface "Local Area Connection" mtu=1482 store=persistent
     Ok.
     ```

     Note
     If you receive an `Element not found` error, replace *`Local Area Connection`* with the `Interface` column output from the To check the MTU setting on a Windows instance (p. 509) procedure that matches your interface.

2. (Optional) Reboot your instance and verify that the MTU setting is correct.

Troubleshooting

If you experience connectivity issues between your EC2 instance and an Amazon Redshift cluster when using jumbo frames, see Queries Appear to Hang in the *Amazon Redshift Cluster Management Guide*

Enabling Enhanced Networking on Windows Instances in a VPC

Amazon EC2 provides enhanced networking capabilities using single root I/O virtualization (SR-IOV) on supported instance types (p. 511). Enabling enhanced networking on your instance results in higher performance (packets per second), lower latency, and lower jitter.

Important
Enhanced networking is already enabled for Windows Server 2012 R2 AMIs. Therefore, if you launch an instance using these AMIs, enhanced networking is already enabled without the need to complete the procedures on this page.

Contents

- Instances that Support Enhanced Networking (p. 511)
- Requirements (p. 511)
- Testing Whether Enhanced Networking Is Enabled (p. 511)
- Enabling Enhanced Networking on Windows (p. 513)

Note that you can get directions for Linux from Enabling Enhanced Networking on Linux Instances in a VPC in the *Amazon EC2 User Guide for Linux Instances*.

Instances that Support Enhanced Networking

The following instances support enhanced networking:

- C3
- C4
- D2
- I2
- M4
- R3

For more information about instance types, see Amazon EC2 Instances.

Requirements

Before enabling enhanced networking, make sure you do the following:

- Launch the instance from a 64-bit English HVM AMI for Windows Server 2012 or Windows Server 2008 R2. (You can't enable enhanced networking on Windows Server 2008 and Windows Server 2003, and enhanced networking is already enabled on Windows Server 2012 R2.)
- Launch the instance using a supported instance type. For more information, see Instances that Support Enhanced Networking (p. 511).
- Launch the instance in a VPC. (You can't enable enhanced networking if the instance is in EC2-Classic.)
- Install and configure either the AWS CLI or Amazon EC2 CLI tools to any computer you choose, preferably your local desktop or laptop. For more information, see Accessing Amazon EC2 (p. 3). If you choose the Amazon EC2 CLI tools, install version 1.6.12.0 or later. You can use the ec2-version command to verify the version of your CLI tools.
- If you have important data on the instance that you want to preserve, you should back that data up now by creating a snapshot. Updating drivers as well as enabling the `sriovNetSupport` attribute may make incompatible instances or operating systems unreachable; if you have a recent backup, your data will still be retained if this happens.

Testing Whether Enhanced Networking Is Enabled

To test whether enhanced networking is already enabled, verify that the driver is installed on your instance and that the `sriovNetSupport` attribute is set.

Driver

To verify that the driver is installed, connect to your instance and open Device Manager. You should see "Intel(R) 82599 Virtual Function" listed under **Network adapters**.

Instance Attribute (sriovNetSupport)

To check whether an instance has the enhanced networking attribute set, use one of the following commands:

- describe-instance-attribute (AWS CLI)

```
C:\> aws ec2 describe-instance-attribute --instance-id instance_id --attribute
 sriovNetSupport
```

If the enhanced networking attribute isn't set, `SriovNetSupport` is empty. Otherwise, it is set as follows:

```
"SriovNetSupport": {
    "Value": "simple"
},
```

- ec2-describe-instance-attribute (Amazon EC2 CLI)

```
C:\> ec2-describe-instance-attribute instance_id --sriov
```

If the enhanced networking attribute isn't set, you'll see no output for this command. Otherwise, the output is as follows:

```
sriovNetSupport instance_id   simple
```

Image Attribute (sriovNetSupport)

To check whether an AMI already has the enhanced networking attribute set, use one of the following commands:

- describe-image-attribute (AWS CLI)

```
C:\> aws ec2 describe-image-attribute --image-id ami_id --attribute sriovNet
Support
```

> **Note**
> This command only works for images that you own. You receive an `AuthFailure` error for images that do not belong to your account.

If the enhanced networking attribute isn't set, `SriovNetSupport` is empty. Otherwise, it is set as follows:

```
"SriovNetSupport": {
    "Value": "simple"
},
```

- ec2-describe-image-attribute (Amazon EC2 CLI)

```
C:\> ec2-describe-image-attribute ami_id --sriov
```

Note
This command only works for images that you own. You will receive an `AuthFailure` error for images that do not belong to your account.

If the enhanced networking attribute isn't set, you'll see no output for this command. Otherwise, the output is as follows:

```
sriovNetSupport ami_id        simple
```

Enabling Enhanced Networking on Windows

If you launched your instance and it does not have enhanced networking enabled already, use the following procedure to enable enhanced networking.

To enable enhanced networking

1. Connect to your instance and log in as the local administrator.
2. From the instance, install the driver as follows:
   a. Download the Intel driver.
   b. In the **Download** folder, locate the `PROWinx64.exe` file. Rename this file `PROWinx64.zip`.
   c. Right-click `PROWinx64.zip` and then click **Extract All**. Specify a destination path and click **Extract**.
   d. Open a Command Prompt window, go to the folder with the extracted files, and run the following command.

      Windows Server 2012

      ```
      C:\> pnputil -a PROXGB\Winx64\NDIS63\vxn63x64.inf
      ```

      Windows Server 2008 R2

      ```
      C:\> pnputil -a PROXGB\Winx64\NDIS62\vxn62x64.inf
      ```

3. From your local computer, stop the instance using the Amazon EC2 console or one of the following commands: stop-instances (AWS CLI) or ec2-stop-instances (Amazon EC2 CLI). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.
4. From a Command Prompt window, enable the enhanced networking attribute using one of the following commands.

   Warning
   There is no way to disable the enhanced networking attribute after you've enabled it.

   - modify-instance-attribute (AWS CLI)

```
C:\> aws ec2 modify-instance-attribute --instance-id instance_id --sriov-net-support simple
```

- ec2-modify-instance-attribute (Amazon EC2 CLI)

```
C:\> ec2-modify-instance-attribute instance_id --sriov simple
```

5. (Optional) Create an AMI from the instance, as described in Creating an Amazon EBS-Backed Windows AMI (p. 68). The AMI inherits the enhanced networking attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking enabled by default.
6. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: start-instances (AWS CLI) or ec2-start-instances (Amazon EC2 CLI). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.

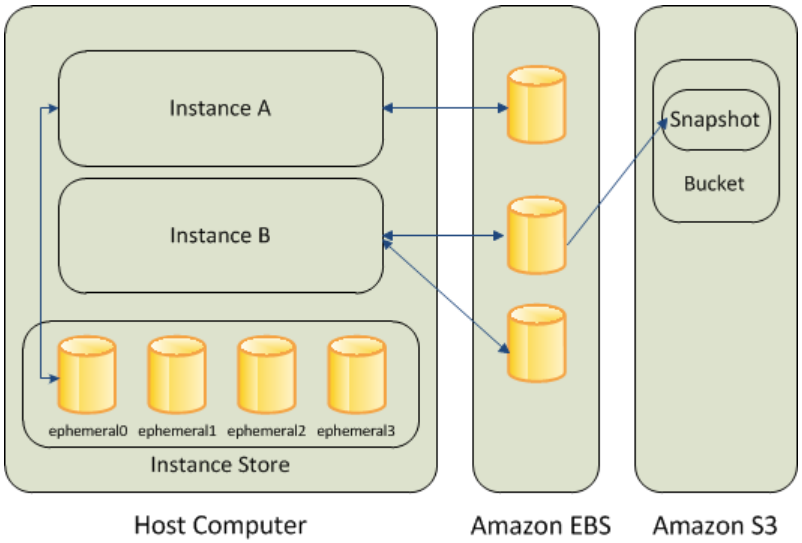
Storage

Amazon EC2 provides you with flexible, cost effective, and easy-to-use data storage options for your instances. Each option has a unique combination of performance and durability. These storage options can be used independently or in combination to suit your requirements.

After reading this section, you should have a good understanding about how you can use the data storage options supported by Amazon EC2 to meet your specific requirements. These storage options include the following:

- Amazon Elastic Block Store (Amazon EBS) (p. 516)
- Amazon EC2 Instance Store (p. 577)
- Amazon Simple Storage Service (Amazon S3) (p. 583)

The following figure shows the relationship between these types of storage.



Amazon EBS

Amazon EBS provides durable, block-level storage volumes that you can attach to a running instance You can use Amazon EBS as a primary storage device for data that requires frequent and granular

updates. For example, Amazon EBS is the recommended storage option when you run a database on an instance.

An EBS volume behaves like a raw, unformatted, external block device that you can attach to a single instance. The volume persists independently from the running life of an instance. After an EBS volume is attached to an instance, you can use it like any other physical hard drive. As illustrated in the previous figure, multiple volumes can be attached to an instance. You can also detach an EBS volume from one instance and attach it to another instance. EBS volumes can also be created as encrypted volumes using the Amazon EBS encryption feature. For more information, see Amazon EBS Encryption (p. 558).

To keep a backup copy of your data, you can create a *snapshot* of an EBS volume, which is stored in Amazon S3. You can create an EBS volume from a snapshot, and attach it to another instance. For more information, see Amazon Elastic Block Store (Amazon EBS) (p. 516).

Amazon EC2 Instance Store

Many instances can access storage from disks that are physically attached to the host computer. This disk storage is referred to as *instance store*. Instance store provides temporary block-level storage for instances. The data on an instance store volume persists only during the life of the associated instance; if you stop or terminate an instance, any data on instance store volumes is lost. For more information, see Amazon EC2 Instance Store (p. 577).

Amazon S3

Amazon S3 is a repository for Internet data. Amazon S3 provides access to reliable and inexpensive data storage infrastructure. It is designed to make web-scale computing easier by enabling you to store and retrieve any amount of data, at any time, from within Amazon EC2 or anywhere on the web. For example, you can use Amazon S3 to store backup copies of your data and applications. For more information, see Amazon Simple Storage Service (Amazon S3) (p. 583).

Adding Storage

Every time you launch an instance from an AMI, a root storage device is created for that instance. The root storage device contains all the information necessary to boot the instance. You can specify storage volumes in addition to the root device volume when you create an AMI or launch an instance using *block device mapping*. For more information, see Block Device Mapping (p. 587).

You can also attach EBS volumes to a running instance. For more information, see Attaching an Amazon EBS Volume to an Instance (p. 529).

Amazon Elastic Block Store (Amazon EBS)

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with EC2 instances. EBS volumes are highly available and reliable storage volumes that can be attached to any running instance that is in the same Availability Zone. EBS volumes that are attached to an EC2 instance are exposed as storage volumes that persist independently from the life of the instance. With Amazon EBS, you pay only for what you use. For more information about Amazon EBS pricing, see the Projecting Costs section of the Amazon Elastic Block Store page.

Amazon EBS is recommended when data changes frequently and requires long-term persistence. EBS volumes are particularly well-suited for use as the primary storage for file systems, databases, or for any applications that require fine granular updates and access to raw, unformatted, block-level storage. Amazon EBS is particularly helpful for database-style applications that frequently encounter many random reads and writes across the data set.

For simplified data encryption, you can launch your EBS volumes as encrypted volumes. Amazon EBS encryption offers you a simple encryption solution for your EBS volumes without the need for you to build,

manage, and secure your own key management infrastructure. When you create an encrypted EBS volume and attach it to a supported instance type, data stored at rest on the volume, disk I/O, and snapshots created from the volume are all encrypted. The encryption occurs on the servers that hosts EC2 instances, providing encryption of data-in-transit from EC2 instances to EBS storage. For more information, see Amazon EBS Encryption (p. 558).

Amazon EBS encryption uses AWS Key Management Service (AWS KMS) master keys when creating encrypted volumes and any snapshots created from your encrypted volumes. The first time you create an encrypted EBS volume in a region, a default master key is created for you automatically. This key is used for Amazon EBS encryption unless you select a Customer Master Key (CMK) that you created separately using the AWS Key Management Service. Creating your own CMK gives you more flexibility, including the ability to create, rotate, disable, define access controls, and audit the encryption keys used to protect your data. For more information, see the AWS Key Management Service Developer Guide.

You can attach multiple volumes to the same instance within the limits specified by your AWS account. Your account has a limit on the number of EBS volumes that you can use, and the total storage available to you. For more information about these limits, and how to request an increase in your limits, see Request to Increase the Amazon EBS Volume Limit.

Contents

- Features of Amazon EBS (p. 517)
- Amazon EBS Volumes (p. 518)
- Amazon EBS Snapshots (p. 549)
- Amazon EBS–Optimized Instances (p. 555)
- Amazon EBS Encryption (p. 558)
- Amazon EBS Volume Performance on Windows Instances (p. 561)
- Amazon EBS Commands (p. 574)

Features of Amazon EBS

- You can create EBS Magnetic volumes from 1 GiB to 1 TiB in size; you can create EBS General Purpose (SSD) and Provisioned IOPS (SSD) volumes up to 16 TiB in size. You can mount these volumes as devices on your Amazon EC2 instances. You can mount multiple volumes on the same instance, but each volume can be attached to only one instance at a time. For more information, see Creating an Amazon EBS Volume (p. 525).
- With General Purpose (SSD) volumes, your volume receives a base performance of 3 IOPS/GiB, with the ability to burst to 3,000 IOPS for extended periods of time. General Purpose (SSD) volumes are ideal for a broad range of use cases such as boot volumes, small and medium size databases, and development and test environments. General Purpose (SSD) volumes support up to 10,000 IOPS and 160 MB/s of throughput. For more information, see General Purpose (SSD) Volumes (p. 521).
- With Provisioned IOPS (SSD) volumes, you can provision a specific level of I/O performance. Provisioned IOPS (SSD) volumes support up to 20,000 IOPS and 320 MB/s of throughput. This allows you to predictably scale to tens of thousands of IOPS per EC2 instance. For more information, see Provisioned IOPS (SSD) Volumes (p. 524).
- EBS volumes behave like raw, unformatted block devices. You can create a file system on top of these volumes, or use them in any other way you would use a block device (like a hard drive). For more information on creating file systems and mounting volumes, see Making an Amazon EBS Volume Available for Use (p. 530).
- You can use encrypted EBS volumes to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications. For more information, see Amazon EBS Encryption (p. 558).
- You can create point-in-time snapshots of EBS volumes, which are persisted to Amazon S3. Snapshots protect data for long-term durability, and they can be used as the starting point for new EBS volumes. The same snapshot can be used to instantiate as many volumes as you wish. These snapshots can be copied across AWS regions. For more information, see Amazon EBS Snapshots (p. 549).

- EBS volumes are created in a specific Availability Zone, and can then be attached to any instances in that same Availability Zone. To make a volume available outside of the Availability Zone, you can create a snapshot and restore that snapshot to a new volume anywhere in that region. You can copy snapshots to other regions and then restore them to new volumes there, making it easier to leverage multiple AWS regions for geographical expansion, data center migration, and disaster recovery. For more information, see Creating an Amazon EBS Snapshot (p. 550), Restoring an Amazon EBS Volume from a Snapshot (p. 527), and Copying an Amazon EBS Snapshot (p. 552).
- A large repository of public data set snapshots can be restored to EBS volumes and seamlessly integrated into AWS cloud-based applications. For more information, see Using Public Data Sets (p. 602).
- Performance metrics, such as bandwidth, throughput, latency, and average queue length, are available through the AWS Management Console. These metrics, provided by Amazon CloudWatch, allow you to monitor the performance of your volumes to make sure that you are providing enough performance for your applications without paying for resources you don't need. For more information, see Amazon EBS Volume Performance on Windows Instances (p. 561).

Amazon EBS Volumes

An Amazon EBS volume is a durable, block-level storage device that you can attach to a single EC2 instance. You can use EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application. EBS volumes persist independently from the running life of an EC2 instance. After a volume is attached to an instance, you can use it like any other physical hard drive. Amazon EBS provides the following volume types: General Purpose (SSD), Provisioned IOPS (SSD), and Magnetic. They differ in performance characteristics and price, allowing you to tailor your storage performance and cost to the needs of your applications. For more information, see Amazon EBS Volume Types (p. 520).

Contents

- Benefits of Using EBS Volumes (p. 518)
- Amazon EBS Volume Types (p. 520)
- Creating an Amazon EBS Volume (p. 525)
- Restoring an Amazon EBS Volume from a Snapshot (p. 527)
- Attaching an Amazon EBS Volume to an Instance (p. 529)
- Making an Amazon EBS Volume Available for Use (p. 530)
- Viewing Volume Information (p. 532)
- Monitoring the Status of Your Volumes (p. 532)
- Detaching an Amazon EBS Volume from an Instance (p. 542)
- Deleting an Amazon EBS Volume (p. 543)
- Expanding the Storage Space of an EBS Volume on Windows (p. 544)

Benefits of Using EBS Volumes

Data Availability

When you create an EBS volume in an Availability Zone, it is automatically replicated within that zone to prevent data loss due to failure of any single hardware component. After you create a volume, you can attach it to any EC2 instance in the same Availability Zone. After you attach a volume, it appears as a native block device similar to a hard drive or other physical device. At that point, the instance can interact with the volume just as it would with a local drive; the instance can format the EBS volume with a file system, such as NTFS, and then install applications.

An EBS volume can be attached to only one instance at a time within the same Availability Zone. However, multiple volumes can be attached to a single instance. If you attach multiple volumes to a device that you have named, you can stripe data across the volumes for increased I/O and throughput performance.

You can get monitoring data for your EBS volumes at no additional charge (this includes data for the root device volumes for EBS-backed instances). For more information, see Monitoring Volumes with CloudWatch (p. 532).

Data Persistence

An EBS volume is off-instance storage that can persist independently from the life of an instance. You continue to pay for the volume usage as long as the data persists.

By default, EBS volumes that are attached to a running instance automatically detach from the instance with their data intact when that instance is terminated. The volume can then be reattached to a new instance, enabling quick recovery. If you are using an EBS-backed instance, you can stop and restart that instance without affecting the data stored in the attached volume. The volume remains attached throughout the stop-start cycle. This enables you to process and store the data on your volume indefinitely, only using the processing and storage resources when required. The data persists on the volume until the volume is deleted explicitly. After a volume is deleted, it can't be attached to any instance. The physical block storage used by deleted EBS volumes is overwritten with zeroes before it is allocated to another account. If you are dealing with sensitive data, you should consider encrypting your data manually or storing the data on a volume that is enabled with Amazon EBS encryption. For more information, see Amazon EBS Encryption (p. 558).

By default, EBS volumes that are created and attached to an instance at launch are deleted when that instance is terminated. You can modify this behavior by changing the value of the flag `DeleteOnTermination` to `false` when you launch the instance. This modified value causes the volume to persist even after the instance is terminated, and enables you to attach the volume to another instance.

Data Encryption

For simplified data encryption, you can create encrypted EBS volumes with the Amazon EBS encryption feature. You can use encrypted EBS volumes to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications. Amazon EBS encryption uses 256-bit Advanced Encryption Standard algorithms (AES-256) and an Amazon-managed key infrastructure. The encryption occurs on the server that hosts the EC2 instance, providing encryption of data-in-transit from the EC2 instance to EBS storage. For more information, see Amazon EBS Encryption (p. 558).

Amazon EBS encryption uses AWS Key Management Service (AWS KMS) master keys when creating encrypted volumes and any snapshots created from your encrypted volumes. The first time you create an encrypted EBS volume in a region, a default master key is created for you automatically. This key is used for Amazon EBS encryption unless you select a Customer Master Key (CMK) that you created separately using the AWS Key Management Service. Creating your own CMK gives you more flexibility, including the ability to create, rotate, disable, define access controls, and audit the encryption keys used to protect your data. For more information, see the AWS Key Management Service Developer Guide.

Snapshots

Amazon EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to Amazon S3, where it is stored redundantly in multiple Availability Zones. The volume does not need be attached to a running instance in order to take a snapshot. As you continue to write data to a volume, you can periodically create a snapshot of the volume to use as a baseline for new volumes. These snapshots can be used to create multiple new EBS volumes, expand the size of a volume, or move volumes across Availability Zones. Snapshots of encrypted EBS volumes are automatically encrypted.

When you create a new volume from a snapshot, it's an exact copy of the original volume at the time the snapshot was taken. EBS volumes that are restored from encrypted snapshots are automatically encrypted.

By optionally specifying a different volume size or a different Availability Zone, you can use this functionality to increase the size of an existing volume or to create duplicate volumes in new Availability Zones. The snapshots can be shared with specific AWS accounts or made public. When you create snapshots, you incur charges in Amazon S3 based on the volume's total size. For a successive snapshot of the volume, you are only charged for any additional data beyond the volume's original size.

Snapshots are incremental backups, meaning that only the blocks on the volume that have changed after your most recent snapshot are saved. If you have a volume with 100 GiB of data, but only 5 GiB of data have changed since your last snapshot, only the 5 GiB of modified data is written to Amazon S3. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.

To help categorize and manage your volumes and snapshots, you can tag them with metadata of your choice. For more information, see Tagging Your Amazon EC2 Resources (p. 609).

Amazon EBS Volume Types

Amazon EBS provides the following volume types, which differ in performance characteristics and price, so that you can tailor your storage performance and cost to the needs of your applications:

- General Purpose (SSD) Volumes (p. 521)
- Provisioned IOPS (SSD) Volumes (p. 524)
- Magnetic Volumes (p. 524)

The following table describes basic use cases and performance characteristics for each volume type.

Characteristic	General Purpose (SSD)	Provisioned IOPS (SSD)	Magnetic

Use cases	• System boot volumes
	• Virtual desktops
	• Small to medium sized databases
	• Development and test environments
		• Critical business applications that require sustained IOPS performance, or more than 10,000 IOPS or 160 MiB/s of throughput per volume
		• Large database workloads, such as:
		  • MongoDB
		  • Microsoft SQL Server
		  • MySQL
		  • PostgreSQL
		  • Oracle
			• Cold workloads where data is infrequently accessed
			• Scenarios where the lowest storage cost is important

Volume size	1 GiB - 16 TiB	4 GiB - 16 TiB	1 GiB - 1 TiB
Maximum throughput	160 MiB/s	320 MiB/s	40-90 MiB/s

Characteristic	General Purpose (SSD)	Provisioned IOPS (SSD)	Magnetic
IOPS performance	Baseline performance of 3 IOPS/GiB (up to 10,000 IOPS) with the ability to burst to 3,000 IOPS for volumes under 1,000 GiB. See I/O Credits and Burst Performance (p. 522)	Consistently performs at provisioned level, up to 20,000 IOPS maximum	Averages 100 IOPS, with the ability to burst to hundreds of IOPS
API and CLI volume name	`gp2`	`io1`	`standard`

Note
The following Amazon EBS volume considerations apply to Windows boot volumes:

- Windows 2003 instances will not boot if the boot volume is 2 TiB (2048 GiB) or greater
- Windows boot volumes must use an MBR partition table, which limits the usable space to 2 TiB, regardless of volume size
- Windows boot volumes 2 TiB (2048 GiB) or greater that have been converted to use a dynamic MBR partition table display an error when examined with Disk Manager

The following Amazon EBS volume considerations apply to Windows data (non-boot) volumes:

- Windows volumes 2 TiB (2048 GiB) or greater must use a GPT partition table to access the entire volume

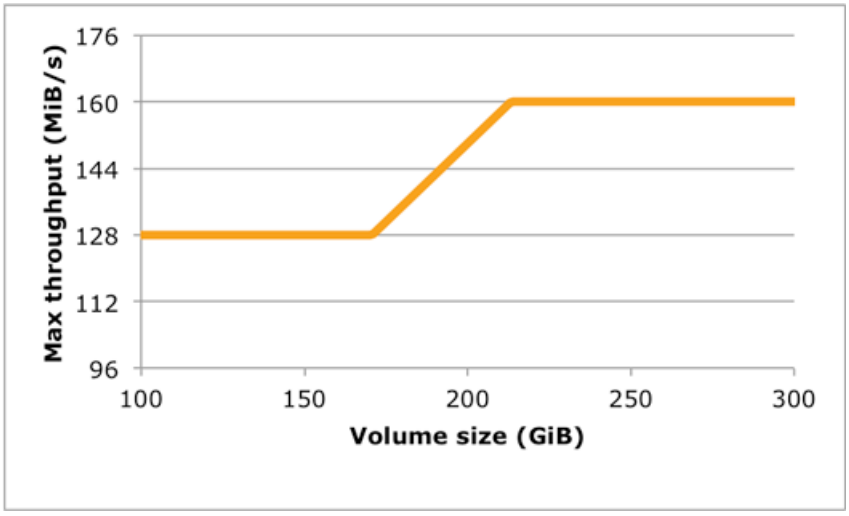
There are several factors that can affect the performance of EBS volumes, such as instance configuration, I/O characteristics, and workload demand. For more information about getting the most out of your EBS volumes, see Amazon EBS Volume Performance on Windows Instances (p. 561).

For detailed pricing information about these volume types, see Amazon EBS Pricing.

General Purpose (SSD) Volumes

General Purpose (SSD) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies, the ability to burst to 3,000 IOPS for extended periods of time, and a baseline performance of 3 IOPS/GiB up to a maximum of 10,000 IOPS (at 3,334 GiB). General Purpose (SSD) volumes can range in size from 1 GiB to 16 TiB.

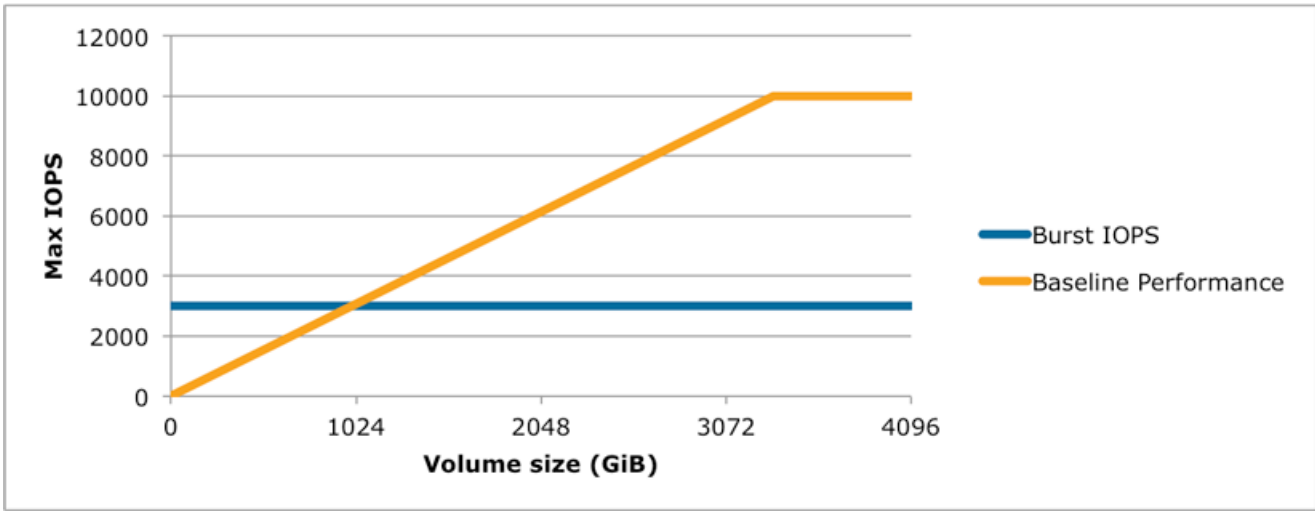
General Purpose (SSD) volumes have a throughput limit range of 128 MiB/s for volumes less than or equal to 170 GiB; for volumes over 170 GiB, this limit increases at the rate of 768 KiB/s per GiB to a maximum of 160 MiB/s (at 214 GiB and larger).



I/O Credits and Burst Performance

General Purpose (SSD) volume performance is governed by volume size, which dictates the baseline performance level of the volume and how quickly it accumulates I/O credits; larger volumes have higher baseline performance levels and accumulate I/O credits faster. I/O credits represent the available bandwidth that your General Purpose (SSD) volume can use to burst large amounts of I/O when more than the baseline performance is needed. The more credits your volume has for I/O, the more time it can burst beyond its baseline performance level and the better it performs when more performance is needed.

Each volume receives an initial I/O credit balance of 5,400,000 I/O credits, which is enough to sustain the maximum burst performance of 3,000 IOPS for 30 minutes. This initial credit balance is designed to provide a fast initial boot cycle for boot volumes and to provide a good bootstrapping experience for other applications. Volumes earn I/O credits every second at a baseline performance rate of 3 IOPS per GiB of volume size. For example, a 100 GiB General Purpose (SSD) volume has a baseline performance of 300 IOPS.



When your volume requires more than the baseline performance I/O level, it simply uses I/O credits in the credit balance to burst to the required performance level, up to a maximum of 3,000 IOPS. Volumes larger than 1,000 GiB have a baseline performance that is equal or greater than the maximum burst performance, and their I/O credit balance never depletes. When your volume uses fewer I/O credits than

it earns in a second, unused I/O credits are added to the I/O credit balance. The maximum I/O credit balance for a volume is equal to the initial credit balance (5,400,000 I/O credits).

The table below lists several volume sizes and the associated baseline performance of the volume (which is also the rate at which it accumulates I/O credits), the burst duration at the 3,000 IOPS maximum (when starting with a full credit balance), and the time in seconds that the volume would take to refill an empty credit balance.

Volume size (GiB)	Baseline performance (IOPS)	Maximum burst duration @ 3,000 IOPS (seconds)	Seconds to fill empty credit balance
1	3	1,802	1,800,000
100	300	2,000	18,000
214 (Min size for max throughput)	642	2,290	15,790
250	750	2,400	7,200
500	1,500	3,600	3,600
750	2,250	7,200	2,400
1,000	3,000	N/A*	N/A*
3,334 (Min size for max IOPS)	10,000	N/A*	N/A*
16,384 (16 TiB, Max volume size)	10,000	N/A*	N/A*

* Bursting and I/O credits are only relevant to volumes under 1,000 GiB, where burst performance exceeds baseline performance.

The burst duration of a volume is dependent on the size of the volume, the burst IOPS required, and the credit balance when the burst begins. This is shown in the equation below:

$$\text{Burst duration} = \frac{\text{(Credit balance)}}{\text{(Burst IOPS) - 3(Volume size in GiB)}}$$

What happens if I empty my I/O credit balance?

If your volume uses all of its I/O credit balance, the maximum IOPS performance of the volume will remain at the baseline IOPS performance level (the rate at which your volume earns credits) and the throughput limit is reduced to the baseline IOPS multiplied by the maximum throughput and divided by 3,000, until I/O demand drops below the baseline level and unused credits are added to the I/O credit balance.

$$\text{Throughput limit with empty I/O credit balance} = \frac{\text{(Max throughput) x (Baseline IOPS)}}{\text{(3,000)}}$$

For example, a 100 GiB volume with an empty credit balance has a baseline IOPS performance limit of 300 IOPS and a throughput limit of 12.8 MiB/s ((128 MiB/s)*300/3000). The larger a volume is, the greater

the baseline performance is and the faster it replenishes the credit balance. For more information on how IOPS are measured, see I/O Characteristics (p. 564).

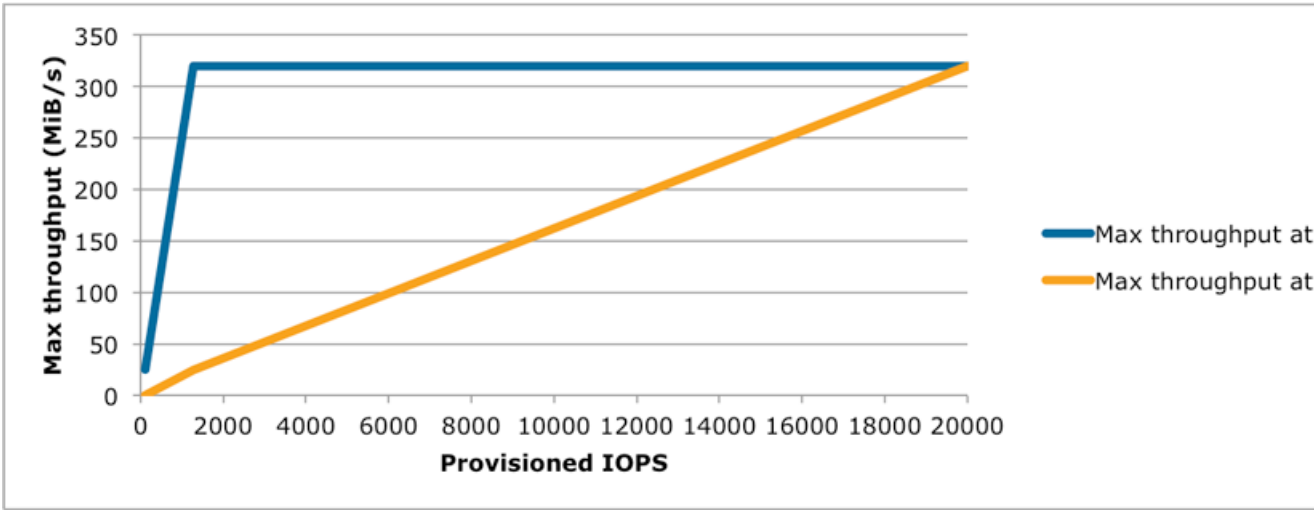
If you notice that your volume performance is frequently limited to the baseline level (due to an empty I/O credit balance), you should consider using a larger General Purpose (SSD) volume (with a higher baseline performance level) or switching to a Provisioned IOPS (SSD) volume for workloads that require sustained IOPS performance greater than 10,000 IOPS.

Provisioned IOPS (SSD) Volumes

Provisioned IOPS (SSD) volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads, that are sensitive to storage performance and consistency in random access I/O throughput. You specify an IOPS rate when you create the volume, and Amazon EBS delivers within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year.

A Provisioned IOPS (SSD) volume can range in size from 4 GiB to 16 TiB and you can provision up to 20,000 IOPS per volume. The ratio of IOPS provisioned to the volume size requested can be a maximum of 30; for example, a volume with 3,000 IOPS must be at least 100 GiB. You can stripe multiple volumes together in a RAID configuration for larger size and greater performance.

Provisioned IOPS (SSD) volumes have a throughput limit range of 256 KiB for each IOPS provisioned, up to a maximum of 320 MiB/s (at 1,280 IOPS).



Your per I/O latency experience depends on the IOPS provisioned and your workload pattern. For the best per I/O latency experience, we recommend you provision an IOPS to GB ratio greater than a 2:1 (for example, a 2,000 IOPS volume would be smaller than 1,000 GiB in this case).

Note
Some AWS accounts created before 2012 might have access to Availability Zones in us-east-1, us-west-1, or ap-northeast-1 that do not support Provisioned IOPS (SSD) volumes. If you are unable to create a Provisioned IOPS (SSD) volume (or launch an instance with a Provisioned IOPS (SSD) volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports Provisioned IOPS (SSD) volumes by creating a 4 GiB Provisioned IOPS (SSD) volume in that zone.

Magnetic Volumes

Magnetic volumes provide the lowest cost per gigabyte of all EBS volume types. Magnetic volumes are backed by magnetic drives and are ideal for workloads performing sequential reads, workloads where data is accessed infrequently, and scenarios where the lowest storage cost is important. These volumes

deliver approximately 100 IOPS on average, with burst capability of up to hundreds of IOPS, and they can range in size from 1 GiB to 1 TiB. Magnetic volumes can be striped together in a RAID configuration for larger size and greater performance.

If you need a greater number of IOPS or higher performance than Magnetic volume can provide, we recommend that you consider General Purpose (SSD) or Provisioned IOPS (SSD) volumes.

Creating an Amazon EBS Volume

You can create an Amazon EBS volume that you can then attach to any EC2 instance within the same Availability Zone. You can choose to create an encrypted EBS volume, but encrypted volumes can only be attached to selected instance types. For more information, see Supported Instance Types (p. 559).

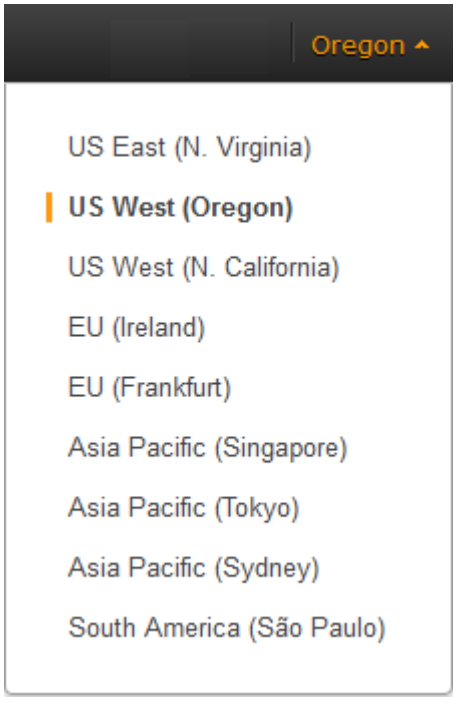
You can also create and attach EBS volumes when you launch instances by specifying a block device mapping. For more information, see Launching an Instance (p. 207) and Block Device Mapping (p. 587). You can restore volumes from previously created snapshots. For more information, see Restoring an Amazon EBS Volume from a Snapshot (p. 527).

If you are creating a volume for a high-performance storage scenario, you should make sure to use a Provisioned IOPS (SSD) volume and attach it to an instance with enough bandwidth to support your application, such as an EBS-optimized instance or an instance with 10 Gigabit network connectivity. For more information, see Amazon EC2 Instance Configuration (p. 562).

When a block of data on a newly created EBS volume is written to for the first time, you might experience longer than normal latency. To avoid the possibility of an increased write latency on a production workload, you should first write to all blocks on the volume to ensure optimal performance; this practice is called pre-warming the volume. For more information, see Pre-Warming Amazon EBS Volumes (p. 565).

To create an EBS volume using the console

1. Open the Amazon EC2 console.
2. From the navigation bar, select the region in which you would like to create your volume. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see Resource Locations (p. 605).



3. Click **Volumes** in the navigation pane.
4. Above the upper pane, click **Create Volume**.
5. In the **Create Volume** dialog box, in the **Volume Type** list, select **General Purpose (SSD)**, **Provisioned IOPS (SSD)** or **Magnetic**. For more information, see Amazon EBS Volume Types (p. 520).

   Note
   Some AWS accounts created before 2012 might have access to Availability Zones in us-east-1, us-west-1, or ap-northeast-1 that do not support Provisioned IOPS (SSD) volumes. If you are unable to create a Provisioned IOPS (SSD) volume (or launch an instance with a Provisioned IOPS (SSD) volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports Provisioned IOPS (SSD) volumes by creating a 4 GiB Provisioned IOPS (SSD) volume in that zone.

6. In the **Size** box, enter the size of the volume, in GiB.

   Note
   The following Amazon EBS volume considerations apply to Windows boot volumes:

   - Windows 2003 instances will not boot if the boot volume is 2 TiB (2048 GiB) or greater
   - Windows boot volumes must use an MBR partition table, which limits the usable space to 2 TiB, regardless of volume size
   - Windows boot volumes 2 TiB (2048 GiB) or greater that have been converted to use a dynamic MBR partition table display an error when examined with Disk Manager

   The following Amazon EBS volume considerations apply to Windows data (non-boot) volumes:

   - Windows volumes 2 TiB (2048 GiB) or greater must use a GPT partition table to access the entire volume

7. For Provisioned IOPS (SSD) volumes, in the **IOPS** box, enter the maximum number of input/output operations per second (IOPS) that the volume should support.
8. In the **Availability Zone** list, select the Availability Zone in which to create the volume.
9. (Optional) To create an encrypted volume, select the **Encrypted** box and choose the master key you want to use when encrypting the volume. You can choose the default master key for your account, or you can choose any Customer Master Key (CMK) that you have previously created using the AWS Key Management Service. Available keys are visible in the **Master Key** drop down menu, or you can paste the full ARN of any key that you have access to. For more information, see the AWS Key Management Service Developer Guide.

   Note
   Encrypted volumes can only be attached to selected instance types. For more information, see Supported Instance Types (p. 559).

10. Click **Yes, Create**.

    Important
    If you receive one of the following errors, the current volume creation would exceed the default storage limit for your account:

```
Maximum number of active volumes bytes, 20, exceeded.
Maximum number of active gp2 volumes bytes, 20, exceeded.
Maximum number of active io1 volumes bytes, 20, exceeded.
```

To view the default service limits for Amazon EBS, see Amazon Elastic Block Store (Amazon EBS) Limits in the *Amazon Web Services General Reference*. To request an increase in your storage limits, see Request to Increase the Amazon EBS Volume Limit.

To create an EBS volume using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- create-volume (AWS CLI)
- ec2-create-volume (Amazon EC2 CLI)
- New-EC2Volume (AWS Tools for Windows PowerShell)

Restoring an Amazon EBS Volume from a Snapshot

You can restore an Amazon EBS volume with data from a snapshot stored in Amazon S3. You need to know the ID of the snapshot you wish to restore your volume from and you need to have access permissions for the snapshot. For more information on snapshots, see Amazon EBS Snapshots (p. 549).

New volumes created from existing Amazon S3 snapshots load lazily in the background. This means that after a volume is created from a snapshot, there is no need to wait for all of the data to transfer from Amazon S3 to your EBS volume before your attached instance can start accessing the volume and all its data. If your instance accesses data that hasn't yet been loaded, the volume immediately downloads the requested data from Amazon S3, and continues loading the rest of the data in the background.

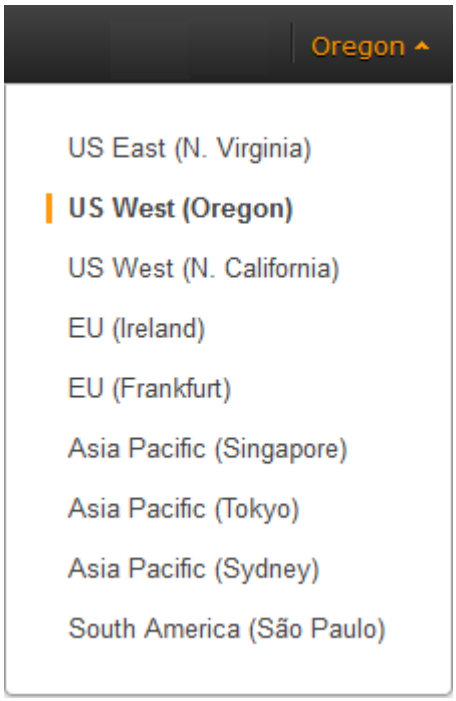
EBS volumes that are restored from encrypted snapshots are automatically encrypted. Encrypted volumes can only be attached to selected instance types. For more information, see Supported Instance Types (p. 559).

When a block of data on a newly restored EBS volume is accessed for the first time, you might experience longer than normal latency. To avoid the possibility of increased read or write latency on a production workload, you should first access all of the blocks on the volume to ensure optimal performance; this practice is called pre-warming the volume. For more information, see Pre-Warming Amazon EBS Volumes (p. 565).

To restore an EBS volume from a snapshot using the console

You can restore your EBS volume from a snapshot using the AWS Management Console as follows.

1. Open the Amazon EC2 console.
2. From the navigation bar, select the region that your snapshot is located in. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see Resource Locations (p. 605). If you need to restore the snapshot to a volume in a different region, you can copy your snapshot to the new region and then restore it to a volume in that region. For more information, see Copying an Amazon EBS Snapshot (p. 552).



3. Click **Volumes** in the navigation pane.
4. Click **Create Volume**.
5. In the **Create Volume** dialog box, in the **Volume Type** list, select **General Purpose (SSD)**, **Provisioned IOPS (SSD)** or **Magnetic**. For more information, see Amazon EBS Volume Types (p. 520).

   Note
   Some AWS accounts created before 2012 might have access to Availability Zones in us-east-1, us-west-1, or ap-northeast-1 that do not support Provisioned IOPS (SSD) volumes. If you are unable to create a Provisioned IOPS (SSD) volume (or launch an instance with a Provisioned IOPS (SSD) volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports Provisioned IOPS (SSD) volumes by creating a 4 GiB Provisioned IOPS (SSD) volume in that zone.

6. In the **Snapshot** field, start typing the ID or description of the snapshot from which you are restoring the volume, and select it from the list of suggested options.

   Note
   Volumes that are restored from encrypted snapshots can only be attached to instances that support Amazon EBS encryption. For more information, see Supported Instance Types (p. 559).

7. In the **Size** box, enter the size of the volume in GiB, or verify the that the default size of the snapshot is adequate.

   If you specify both a volume size and a snapshot ID, the size must be equal to or greater than the snapshot size. When you select a volume type and a snapshot ID, minimum and maximum sizes for the volume are shown next to the **Size** list. Any AWS Marketplace product codes from the snapshot are propagated to the volume.

   Note
   The following Amazon EBS volume considerations apply to Windows boot volumes:

   - Windows 2003 instances will not boot if the boot volume is 2 TiB (2048 GiB) or greater
   - Windows boot volumes must use an MBR partition table, which limits the usable space to 2 TiB, regardless of volume size

- Windows boot volumes 2 TiB (2048 GiB) or greater that have been converted to use a dynamic MBR partition table display an error when examined with Disk Manager

The following Amazon EBS volume considerations apply to Windows data (non-boot) volumes:

- Windows volumes 2 TiB (2048 GiB) or greater must use a GPT partition table to access the entire volume

8. For Provisioned IOPS (SSD) volumes, in the **IOPS** box, enter the maximum number of input/output operations per second (IOPS) that the volume can support.
9. In the **Availability Zone** list, select the Availability Zone in which to create the volume. EBS volumes can only be attached to EC2 instances within the same Availability Zone.
10. Click **Yes, Create**.

Important
If you restored a snapshot to a larger volume than the default for that snapshot, you need to extend the file system on the volume to take advantage of the extra space. For more information, see Expanding the Storage Space of an EBS Volume on Windows (p. 544).

To restore an EBS volume using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- create-volume (AWS CLI)
- ec2-create-volume (Amazon EC2 CLI)
- New-EC2Volume (AWS Tools for Windows PowerShell)

Attaching an Amazon EBS Volume to an Instance

You can attach an EBS volumes to one of your instances that is in the same Availability Zone as the volume.

Prerequisites

- Determine the device names that you'll use. For more information, see Device Naming on Windows Instances (p. 586).
- Determine how many volumes you can attach to your instance. For more information, see Instance Volume Limits (p. 585).
- If a volume is encrypted, it can only be attached to an instance that supports Amazon EBS encryption. For more information, see Supported Instance Types (p. 559).
- If a volume has an AWS Marketplace product code:
  - The volume can only be attached to a stopped instance.
  - You must be subscribed to the AWS Marketplace code that is on the volume.
  - The configuration (instance type, operating system) of the instance must support that specific AWS Marketplace code. For example, you cannot take a volume from a Windows instance and attach it to a Linux instance.
  - AWS Marketplace product codes are copied from the volume to the instance.

To attach an EBS volume to an instance using the console

1. Open the Amazon EC2 console.
2. Click **Volumes** in the navigation pane.
3. Select a volume and then click **Attach Volume**.
4. In the **Attach Volume** dialog box, start typing the name or ID of the instance to attach the volume to in the **Instance** box, and select it from the list of suggestion options (only instances that are in the same Availability Zone as the volume are displayed).
5. You can keep the suggested device name, or enter a different supported device name.

   Important
   The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different from the name that Amazon EC2 recommends.
6. Click **Attach**.
7. Connect to your instance and make the volume available. For more information, see Making an Amazon EBS Volume Available for Use (p. 530).

To attach an EBS volume to an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- attach-volume (AWS CLI)
- ec2-attach-volume (Amazon EC2 CLI)
- Add-EC2Volume (AWS Tools for Windows PowerShell)

Making an Amazon EBS Volume Available for Use

After you attach an Amazon EBS volume to your instance, it is exposed as a block device. You can format the volume with any file system and then mount it. After you make the EBS volume available for use, you can access it in the same ways that you access any other volume. Any data written to this file system is written to the EBS volume and is transparent to applications using the device.

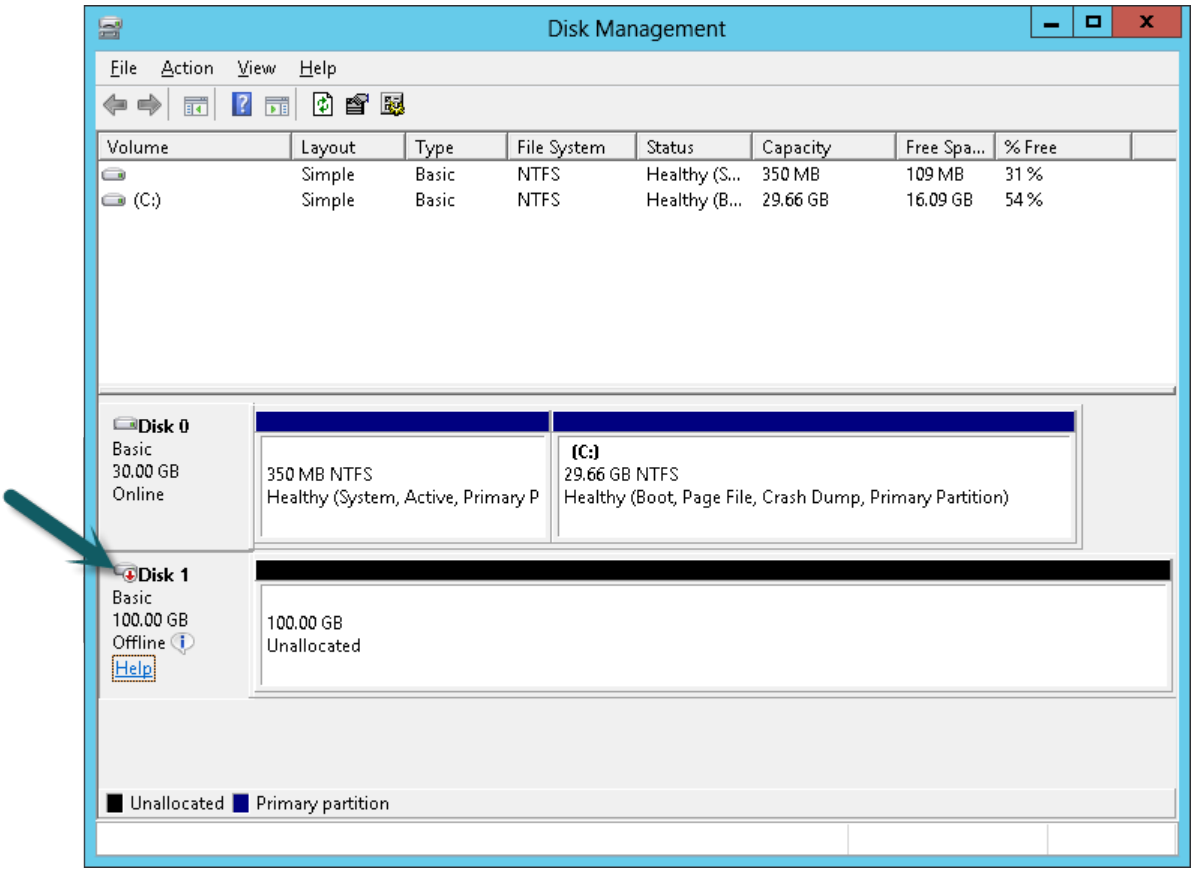
Note that you can take snapshots of your EBS volume for backup purposes or to use as a baseline when you create another volume. For more information, see Amazon EBS Snapshots (p. 549).

Making the Volume Available on Windows

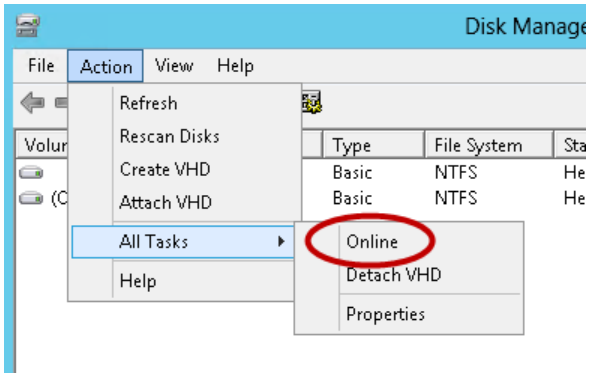
Use the following procedure to make the volume available. Note that you can get directions for volumes on a Linux instance from Making the Volume Available on Linux in the *Amazon EC2 User Guide for Linux Instances*.

To use an EBS volume

1. Log in to your Windows instance using Remote Desktop. For more information, see, Connecting to Your Windows Instance Using RDP (p. 216).
2. Start the Disk Management utility. On Windows Server 2012, on the taskbar, right-click the Windows logo, and then select **Disk Management**. On Windows Server 2008, click **Start**, point to **Administrative Tools**, select **Computer Management**, and then select **Disk Management**.
3. Select the disk that represents the new EBS volume.



4. On the **Disk Management** menu, select **Action** - **All Tasks** - **Online**.



5. (Conditional) A new disk needs to be initialized before it can be used.

   Caution
   If you're mounting a volume that already has data on it (for example, a public data set), make sure that you don't reformat the volume and delete the existing data.

   To initialize a new disk:

   a. In the Disk Management utility, select the new EBS volume disk.
   b. On the **Disk Management** menu, select **Action** - **All Tasks** - **Initialize Disk**.

c. In the **Initialize Disk** dialog, select the disk to initialize, select the desired partition style, and press **OK**.

Viewing Volume Information

You can view descriptive information for your Amazon EBS volumes in a selected region at a time in the AWS Management Console. You can also view detailed information about a single volume, including the size, volume type, whether or not the volume is encrypted, which master key was used to encrypt the volume, and the specific instance to which the volume is attached.

To view information about an EBS volume using the console

1. Open the Amazon EC2 console.
2. Click **Volumes** in the navigation pane.
3. To view more information about a volume, select it.

To view information about an EBS volume using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-volumes (AWS CLI)
- ec2-describe-volumes (Amazon EC2 CLI)
- Get-EC2Volume (AWS Tools for Windows PowerShell)

Monitoring the Status of Your Volumes

Amazon Web Services (AWS) automatically provides data, such as Amazon CloudWatch metrics and volume status checks, that you can use to monitor your Amazon Elastic Block Store (Amazon EBS) volumes.

Contents

- Monitoring Volumes with CloudWatch (p. 532)
- Monitoring Volumes with Status Checks (p. 535)
- Monitoring Volume Events (p. 537)
- Working with an Impaired Volume (p. 538)
- Working with the AutoEnableIO Volume Attribute (p. 541)

Monitoring Volumes with CloudWatch

CloudWatch metrics are statistical data that you can use to view, analyze, and set alarms on the operational behavior of your volumes.

The following table describes the types of monitoring data available for your Amazon EBS volumes.

Type	Description
Basic	Data is available automatically in 5-minute periods at no charge. This includes data for the root device volumes for Amazon EBS-backed instances.

Type	Description
Detailed	Provisioned IOPS (SSD) volumes automatically send one-minute metrics to CloudWatch.

When you get data from CloudWatch, you can include a `Period` request parameter to specify the granularity of the returned data. This is different than the period that we use when we collect the data (5-minute periods). We recommend that you specify a period in your request that is equal to or larger than the collection period to ensure that the returned data is valid.

You can get the data using either the Amazon CloudWatch API or the Amazon EC2 console. The console takes the raw data from the Amazon CloudWatch API and displays a series of graphs based on the data. Depending on your needs, you might prefer to use either the data from the API or the graphs in the console.

Amazon EBS Metrics

You can use the Amazon CloudWatch `GetMetricStatistics` API to get any of the Amazon EBS volume metrics listed in the following table. Similar metrics are grouped together in the table, and the metrics in the first two rows are also available for the local stores on Amazon EC2 instances.

Metric	Description

`VolumeReadBytes`
 	
`VolumeWriteBytes`
	Provides information on the I/O operations in a specified period of time. The `Sum` statistic reports the total number of bytes transferred during the period. The `Average` statistic reports the average size of each I/O operation during the period. The `SampleCount` statistic reports the total number of I/O operations during the period. The `Minimum` and `Maximum` statistics are not relevant for this metric. Data is only reported to Amazon CloudWatch when the volume is active. If the volume is idle, no data is reported to Amazon CloudWatch.
	
	Units: Bytes

`VolumeReadOps`
 	
`VolumeWriteOps`
	The total number of I/O operations in a specified period of time.
	
	Note
	To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period.
	
	Units: Count

`VolumeTotalReadTime`
 	
`VolumeTotalWriteTime`
	The total number of seconds spent by all operations that completed in a specified period of time. If multiple requests are submitted at the same time, this total could be greater than the length of the period. For example, for a period of 5 minutes (300 seconds): if 700 operations completed during that period, and each operation took 1 second, the value would be 700 seconds.
	
	Units: Seconds

`VolumeIdleTime`	The total number of seconds in a specified period of time when no read or write operations were submitted. Units: Seconds

Metric	Description
`VolumeQueueLength`	The number of read and write operation requests waiting to be completed in a specified period of time. Units: Count
`VolumeThroughputPercentage`	Used with Provisioned IOPS (SSD) volumes only. The percentage of I/O operations per second (IOPS) delivered of the total IOPS provisioned for an Amazon EBS volume. Provisioned IOPS (SSD) volumes deliver within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year. **Note** During a write, if there are no other pending I/O requests in a minute, the metric value will be 100 percent. Also, a volume's I/O performance may become degraded temporarily due to an action you have taken (e.g., creating a snapshot of a volume during peak usage, running the volume on a non-EBS-optimized instance, accessing data on the volume for the first time). Units: Percent
`VolumeConsumedReadWriteOps`	Used with Provisioned IOPS (SSD) volumes only. The total amount of read and write operations (normalized to 256K capacity units) consumed in a specified period of time. I/O operations that are smaller than 256K each count as 1 consumed IOPS. I/O operations that are larger than 256K are counted in 256K capacity units. For example, a 1024K I/O would count as 4 consumed IOPS. Units: Count

Graphs in the Amazon EC2 console

After you create a volume, you can go to the Amazon EC2 console and view the volume's monitoring graphs. They're displayed when you select the volume on the **Volumes** page in the EC2 console. A **Monitoring** tab is displayed next to the volume's **Description** tab. The following table lists the graphs that are displayed. The column on the right describes how the raw data metrics from the Amazon CloudWatch API are used to produce each graph. The period for all the graphs is 5 minutes.

Graph Name	Description Using Raw Metrics
Read Bandwidth (KiB/s)	Sum(VolumeReadBytes) / Period / 1024
Write Bandwidth (KiB/s)	Sum(VolumeWriteBytes) / Period / 1024
Read Throughput (Ops/s)	Sum(VolumeReadOps) / Period
Write Throughput (Ops/s)	Sum(VolumeWriteOps) / Period
Avg Queue Length (ops)	Avg(VolumeQueueLength)
% Time Spent Idle	Sum(VolumeIdleTime) / Period * 100
Avg Read Size (KiB/op)	Avg(VolumeReadBytes) / 1024
Avg Write Size (KiB/op)	Avg(VolumeWriteBytes) / 1024

Graph Name	Description Using Raw Metrics
Avg Read Latency (ms/op)	Avg(VolumeTotalReadTime) * 1000
Avg Write Latency (ms/op)	Avg(VolumeTotalWriteTime) * 1000

For the average latency graphs and average size graphs, the average is calculated over the total number of operations (read or write, whichever is applicable to the graph) that completed during the period.

The AWS Management Console contains a console for Amazon CloudWatch. In the Amazon CloudWatch console you can search and browse all your AWS resource metrics, view graphs to troubleshoot issues and discover trends, create and edit alarms to be notified of problems, and see at-a-glance overviews of your alarms and AWS resources. For more information, see AWS Management Console in the *Amazon CloudWatch Developer Guide*.

Monitoring Volumes with Status Checks

Volume status checks enable you to better understand, track, and manage potential inconsistencies in the data on an Amazon EBS volume. They are designed to provide you with the information that you need to determine whether your Amazon EBS volumes are impaired, and to help you control how a potentially inconsistent volume is handled.

Volume status checks are automated tests that run every 5 minutes and return a pass or fail status. If all checks pass, the status of the volume is `ok`. If a check fails, the status of the volume is `impaired`. If the status is `insufficient-data`, the checks may still be in progress on the volume. You can view the results of volume status checks to identify any impaired volumes and take any necessary actions.

When Amazon EBS determines that a volume's data is potentially inconsistent, the default is that it disables I/O to the volume from any attached EC2 instances, which helps to prevent data corruption. After I/O is disabled, the next volume status check fails, and the volume status is `impaired`. In addition, you'll see an event that lets you know that I/O is disabled, and that you can resolve the impaired status of the volume by enabling I/O to the volume. We wait until you enable I/O to give you the opportunity to decide whether to continue to let your instances use the volume, or to run a consistency check using a command, such as **fsck** (Linux) or **chkdsk** (Windows), before doing so.

> **Note**
> Volume status is based on the volume status checks, and does not reflect the volume state. Therefore, volume status does not indicate volumes in the `error` state (for example, when a volume is incapable of accepting I/O.)

If the consistency of a particular volume is not a concern for you, and you'd prefer that the volume be made available immediately if it's impaired, you can override the default behavior by configuring the volume to automatically enable I/O. If you enable the `AutoEnableIO` volume attribute, the volume status check continues to pass. In addition, you'll see an event that lets you know that the volume was determined to be potentially inconsistent, but that its I/O was automatically enabled. This enables you to check the volume's consistency or replace it at a later time.

The I/O performance status check compares actual volume performance to the expected performance of a volume and alerts you if the volume is performing below expectations. This status check is only available for Provisioned IOPS (SSD) volumes that are attached to an instance and is not valid for General Purpose (SSD) and Magnetic volumes. The I/O performance status check is performed once every minute and CloudWatch collects this data every 5 minutes, so it may take up to 5 minutes from the moment you attach a Provisioned IOPS (SSD) volume to an instance for this check to report the I/O performance status.

> **Important**
> While pre-warming Provisioned IOPS (SSD) volumes that were restored from snapshots, the performance of the volume may drop below 50 percent of its expected level, which causes the

volume to display a `warning` state in the **I/O Performance** status check. This is expected, and you can ignore the `warning` state on Provisioned IOPS (SSD) volumes while you are pre-warming them. For more information, see Pre-Warming Amazon EBS Volumes (p. 565).

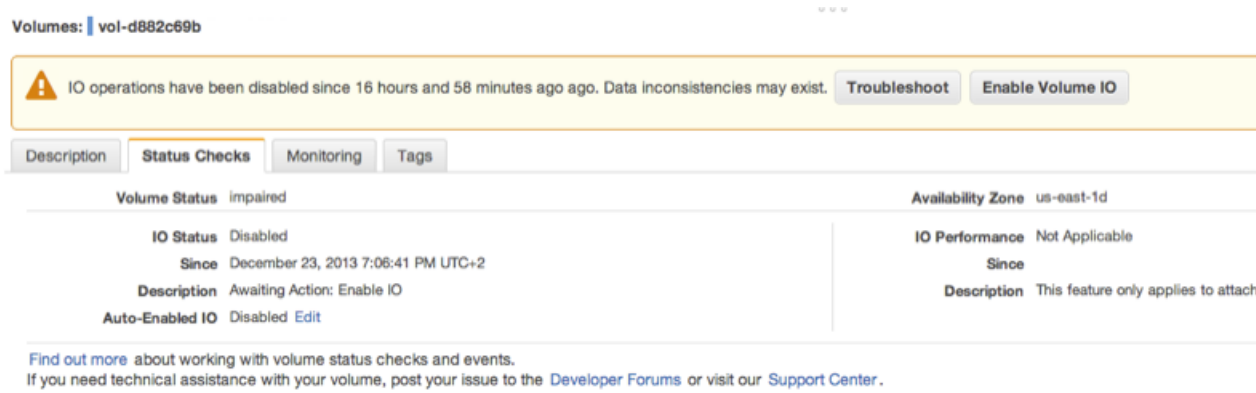
The following table lists statuses for Amazon EBS volumes.

Volume Status	I/O Enabled Status	I/O Performance Status (only available for Provisioned IOPS volumes)
`ok`	Enabled (I/O Enabled or I/O Auto-Enabled)	Normal (Volume performance is as expected)
`warning`	Enabled (I/O Enabled or I/O Auto-Enabled)	Degraded (Volume performance is below expectations) Severely Degraded (Volume performance is well below expectations)
`impaired`	Enabled (I/O Enabled or I/O Auto-Enabled) Disabled (Volume is offline and pending recovery, or is waiting for the user to enable I/O)	Stalled (Volume performance is severely impacted) Not Available (Unable to determine I/O performance because I/O is disabled)
`insufficient-data`	Enabled (I/O Enabled or I/O Auto-Enabled) Insufficient Data	Insufficient Data

To view and work with status checks, you can use the Amazon EC2 console, the API, or the command line interface.

To view status checks in the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Volumes**.
3. On the **EBS Volumes** page, the **Volume Status** column lists the operational status of each volume.
4. To view an individual volume's status, select the volume, and then click the **Status Checks** tab.



5. If you have a volume with a failed status check (status is `impaired`), see Working with an Impaired Volume (p. 538).

Alternatively, you can use the **Events** pane to view all events for your instances and volumes in a single pane. For more information, see Monitoring Volume Events (p. 537).

To view volume status information with the command line

You can use one of the following commands to view the status of your Amazon EBS volumes. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-volume-status (AWS CLI)
- ec2-describe-volume-status (Amazon EC2 CLI)
- Get-EC2VolumeStatus (AWS Tools for Windows PowerShell)

Monitoring Volume Events

When Amazon EBS determines that a volume's data is potentially inconsistent, it disables I/O to the volume from any attached EC2 instances by default. This causes the volume status check to fail, and creates a volume status event that indicates the cause of the failure.

To automatically enable I/O on a volume with potential data inconsistencies, change the setting of the `AutoEnableIO` volume attribute. For more information about changing this attribute, see Working with an Impaired Volume (p. 538).

Each event includes a start time that indicates the time at which the event occurred, and a duration that indicates how long I/O for the volume was disabled. The end time is added to the event when I/O for the volume is enabled.

Volume status events include one of the following descriptions:

Awaiting Action: Enable IO
: Volume data is potentially inconsistent. I/O is disabled for the volume until you explicitly enable it. The event description changes to **IO Enabled** after you explicitly enable I/O.

IO Enabled
: I/O operations were explicitly enabled for this volume.

IO Auto-Enabled
: I/O operations were automatically enabled on this volume after an event occurred. We recommend that you check for data inconsistencies before continuing to use the data.

Normal
: For Provisioned IOPS (SSD) volumes only. Volume performance is as expected.

Degraded
: For Provisioned IOPS (SSD) volumes only. Volume performance is below expectations.

Severely Degraded
: For Provisioned IOPS (SSD) volumes only. Volume performance is well below expectations.

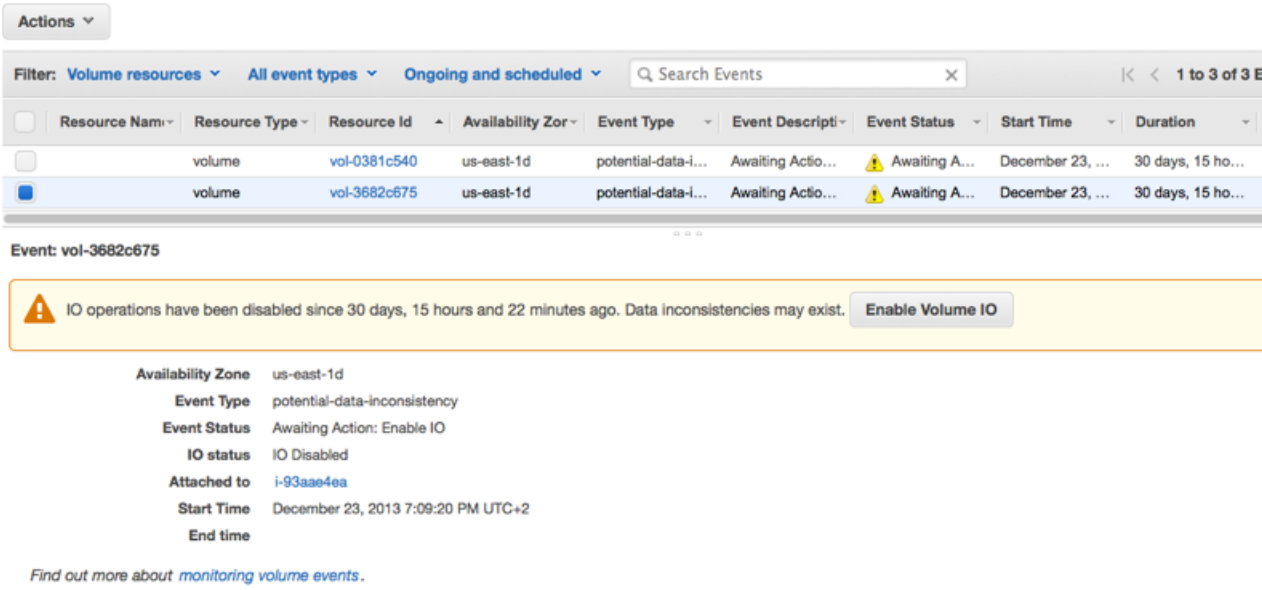
Stalled
: For Provisioned IOPS (SSD) volumes only. Volume performance is severely impacted.

You can view events for your volumes using the Amazon EC2 console, the API, or the command line interface.

To view events for your volumes in the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.

2. In the navigation pane, click **Events**.
3. All instances and volumes that have events are listed. You can filter by volume to view only volume status. You can also filter on specific status types.
4. Select a volume to view its specific event.



If you have a volume where I/O is disabled, see Working with an Impaired Volume (p. 538). If you have a volume where I/O performance is below normal, this might be a temporary condition due to an action you have taken (e.g., creating a snapshot of a volume during peak usage, running the volume on an instance that cannot support the I/O bandwidth required, accessing data on the volume for the first time, etc.).

To view events for your volumes with the command line

You can use one of the following commands to view event information for your Amazon EBS volumes. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-volume-status (AWS CLI)
- ec2-describe-volume-status (Amazon EC2 CLI)
- Get-EC2VolumeStatus (AWS Tools for Windows PowerShell)

Working with an Impaired Volume

This section discusses your options if a volume is impaired because the volume's data is potentially inconsistent.

Options

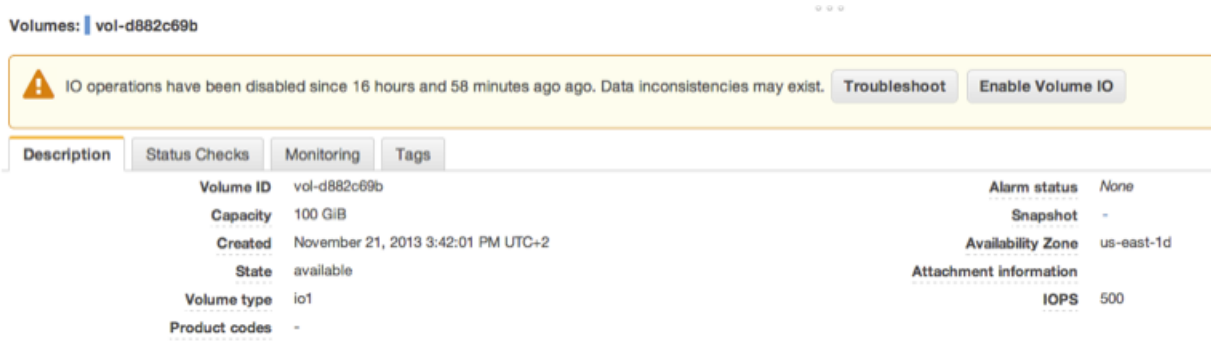
- Option 1: Perform a Consistency Check on the Volume Attached to its Instance (p. 539)
- Option 2: Perform a Consistency Check on the Volume Using Another Instance (p. 539)
- Option 3: Delete the Volume If You No Longer Need It (p. 540)

Option 1: Perform a Consistency Check on the Volume Attached to its Instance

The simplest option is to enable I/O and then perform a data consistency check on the volume while the volume is still attached to its Amazon EC2 instance.

To perform a consistency check on an attached volume

1. Stop any applications from using the volume.
2. Enable I/O on the volume.
   a. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
   b. In the navigation pane, click **Volumes**.
   c. Select the volume on which you want to enable I/O operations.
   d. In the details pane, click **Enable Volume IO**.



   e. In **Enable Volume IO**, click **Yes, Enable**.

3. Check the data on the volume.
   a. Run the **fsck** (Linux) or **chkdsk** (Windows) command.
   b. (Optional) Review any available application or system logs for relevant error messages.
   c. If the volume has been impaired for more than 20 minutes you can contact support. Click **Troubleshoot**, and then on the **Troubleshoot Status Checks** dialog box, click **Contact Support** to submit a support case.

For information about using the command line interface to enable I/O for a volume, see ec2-enable-volume-io in the *Amazon EC2 Command Line Reference*. For information about using the API to enable I/O for a volume, see EnableVolumeIO in the *Amazon EC2 API Reference*.

Option 2: Perform a Consistency Check on the Volume Using Another Instance

Use the following procedure to check the volume outside your production environment.

> **Important**
> This procedure may cause the loss of write I/Os that were suspended when volume I/O was disabled.

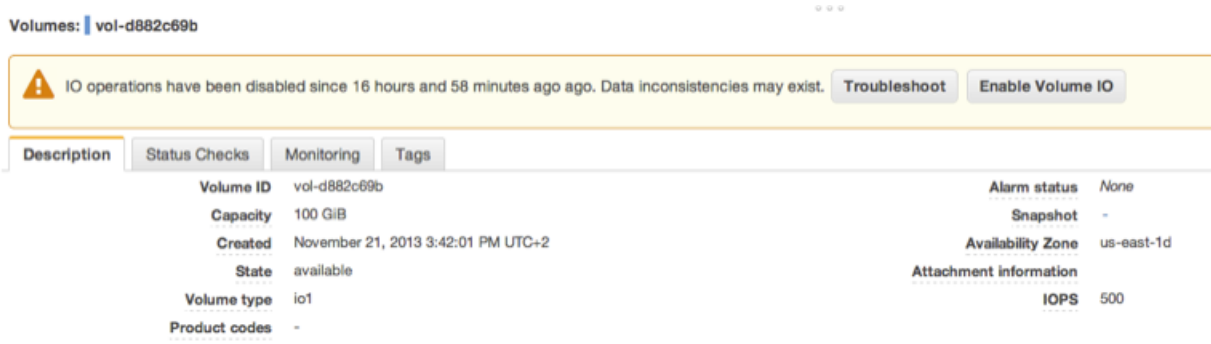
To perform a consistency check on a volume in isolation

1. Stop any applications from using the volume.
2. Detach the volume from the instance.

a. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
b. In the navigation pane, click **Volumes**.
c. Select the volume that you want to detach.
d. Click **Actions**, and then click **Force Detach Volume**. You'll be prompted for confirmation.

3. Enable I/O on the volume.

a. In the navigation pane, click **Volumes**.
b. Select the volume that you detached in the previous step.
c. In the details pane, click **Enable Volume IO**.



d. In the **Enable Volume IO** dialog box, click **Yes, Enable**.

4. Attach the volume to another instance. For information, see Launch Your Instance (p. 206) and Attaching an Amazon EBS Volume to an Instance (p. 529).
5. Check the data on the volume.

a. Run the **fsck** (Linux) or **chkdsk** (Windows) command.
b. (Optional) Review any available application or system logs for relevant error messages.
c. If the volume has been impaired for more than 20 minutes, you can contact support. Click **Troubleshoot**, and then in the troubleshooting dialog box, click **Contact Support** to submit a support case.

For information about using the command line interface to enable I/O for a volume, see ec2-enable-volume-io in the *Amazon EC2 Command Line Reference*. For information about using the API to enable I/O for a volume, see EnableVolumeIO in the *Amazon EC2 API Reference*.

Option 3: Delete the Volume If You No Longer Need It

If you want to remove the volume from your environment, simply delete it. For information about deleting a volume, see Deleting an Amazon EBS Volume (p. 543).

If you have a recent snapshot that backs up the data on the volume, you can create a new volume from the snapshot. For information about creating a volume from a snapshot, see Restoring an Amazon EBS Volume from a Snapshot (p. 527).

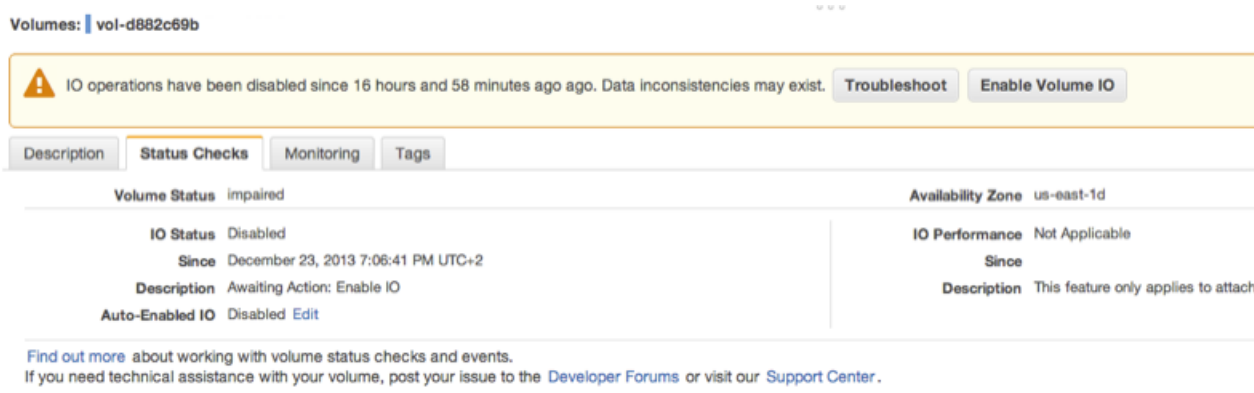
Working with the AutoEnableIO Volume Attribute

When Amazon EBS determines that a volume's data is potentially inconsistent, it disables I/O to the volume from any attached EC2 instances by default. This causes the volume status check to fail, and creates a volume status event that indicates the cause of the failure. If the consistency of a particular volume is not a concern, and you prefer that the volume be made available immediately if it's `impaired`, you can override the default behavior by configuring the volume to automatically enable I/O. If you enable the `AutoEnableIO` volume attribute, I/O between the volume and the instance is automatically reenabled and the volume's status check will pass. In addition, you'll see an event that lets you know that the volume was in a potentially inconsistent state, but that its I/O was automatically enabled. When this event occurs, you should check the volume's consistency and replace it if necessary. For more information, see Monitoring Volume Events (p. 537).

This section explains how to view and modify the `AutoEnableIO` attribute of a volume using the Amazon EC2 console, the command line interface, or the API.

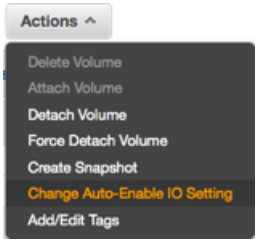
To view the AutoEnableIO attribute of a volume in the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Volumes**.
3. Select the volume.
4. In the lower pane, click the **Status Checks** tab.
5. In the **Status Checks** tab, **Auto-Enable IO** displays the current setting for your volume, either `Enabled` or `Disabled`.

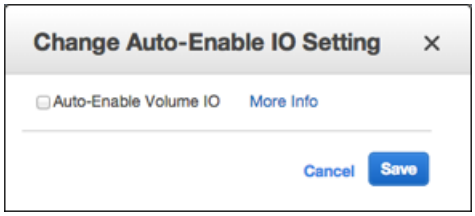


To modify the AutoEnableIO attribute of a volume in the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Volumes**.
3. Select the volume.
4. At the top of the **Volumes** page, click **Actions**.
5. Click **Change Auto-Enable IO Setting**.



6. In the **Change Auto-Enable IO Setting** dialog box, select the **Auto-Enable Volume IO** option to automatically enable I/O for an impaired volume. To disable the feature, clear the option.



7. Click **Save**.

Alternatively, instead of completing steps 4-6 in the previous procedure, go to the **Status Checks** tab and click **Edit**.

To view or modify the AutoEnableIO attribute of a volume with the command line

You can use one of the following commands to view the `AutoEnableIO` attribute of your Amazon EBS volumes. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-volume-attribute (AWS CLI)
- ec2-describe-volume-attribute (Amazon EC2 CLI)
- Get-EC2VolumeAttribute (AWS Tools for Windows PowerShell)

To modify the `AutoEnableIO` attribute of a volume, you can use one of the commands below.

- modify-volume-attribute (AWS CLI)
- ec2-modify-volume-attribute (Amazon EC2 CLI)
- Edit-EC2VolumeAttribute (AWS Tools for Windows PowerShell)

Detaching an Amazon EBS Volume from an Instance

You can detach an Amazon EBS volume from an instance explicitly or by terminating the instance. However, if the instance that the volume is attached to is running, you must unmount the volume (from the instance) before you detach it. Failure to do so results in the volume being stuck in the busy state while it is trying to detach, which could possibly damage the file system or the data it contains.

If an EBS volume is the root device of an instance, you must stop the instance before you can detach the volume.

When a volume with an AWS Marketplace product code is detached from an instance, the product code is no longer associated with the instance.

Important
After you detach a volume, you are still charged for volume storage as long as the storage amount exceeds the limit of the Free Usage Tier. You must delete a volume to avoid incurring further charges. For more information, see Deleting an Amazon EBS Volume (p. 543).

This example unmounts the volume and then explicitly detaches it from the instance. This is useful when you want to terminate an instance or attach a volume to a different instance. To verify that the volume is no longer attached to the instance, see Viewing Volume Information (p. 532).

Note that you can reattach a volume that you detached (without unmounting it), but it might not get the same mount point and the data on the volume might be out of sync if there were writes to the volume in progress when it was detached.

To detach an EBS volume using the console

1. First, unmount the volume. Open **Disk Management**, right-click the volume, and then select **Change Drive Letter and Path**. Select the mount point and then click **Remove**.
2. Open the Amazon EC2 console.
3. Click **Volumes** in the navigation pane.
4. Select a volume and then click **Detach Volume**.
5. In the confirmation dialog box, click **Yes, Detach**.

To detach an EBS volume from an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- detach-volume (AWS CLI)
- ec2-detach-volume (Amazon EC2 CLI)
- Dismount-EC2Volume (AWS Tools for Windows PowerShell)

Troubleshooting

If your volume stays in the *detaching* state, you can force the detachment by clicking **Force Detach**. Forcing the detachment can lead to data loss or a corrupted file system. Use this option only as a last resort to detach a volume from a failed instance, or if you are detaching a volume with the intention of deleting it. The instance doesn't get an opportunity to flush file system caches or file system metadata. If you use this option, you must perform file system check and repair procedures.

If you've tried to force the volume to detach multiple times over several minutes and it stays in the *detaching* state, you can post a request for help to the Amazon EC2 forum. To help expedite a resolution, include the volume ID and describe the steps that you've already taken.

Deleting an Amazon EBS Volume

After you no longer need an Amazon EBS volume, you can delete it. After deletion, its data is gone and the volume can't be attached to any instance. However, before deletion, you can store a snapshot of the volume, which you can use to recreate the volume later.

To delete an EBS volume using the console

1. Open the Amazon EC2 console.
2. Click **Volumes** in the navigation pane.
3. Select a volume and click **Delete Volume**.
4. In the confirmation dialog box, click **Yes, Delete**.

To delete an EBS volume using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- delete-volume (AWS CLI)
- ec2-delete-volume (Amazon EC2 CLI)
- Remove-EC2Volume (AWS Tools for Windows PowerShell)

Expanding the Storage Space of an EBS Volume on Windows

You can increase the storage space of an existing EBS volume without losing the data on the volume. To do this, you migrate your data to a larger volume and then extend the file system on the volume to recognize the newly-available space. After you verify that your new volume is working properly, you can delete the old volume.

Tasks

- Migrating Your Data to a Larger Volume (p. 544)
- Extending a Windows File System (p. 546)
- Deleting the Old Volume (p. 549)

If you need to expand the storage space of a volume on a Linux instance, see Expanding the Storage Space of a Volume in the *Amazon EC2 User Guide for Linux Instances*.

If you create a larger volume, you will be charged for the additional storage. For more information, see the *Amazon Elastic Block Store* section on the Amazon EC2 Pricing page.

Migrating Your Data to a Larger Volume

To migrate your data to a larger volume

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and then locate the instance with the volume that you want to expand.
3. Make a note of the instance ID and Availability Zone. You will specify this information when you attach a new volume to the instance later in this topic.
4. Verify that the instance **Shutdown Behavior** is set to **Stop** and not **Terminate**.

   a. Choose the instance.
   b. From the context-menu (right-click) choose **Instance Settings**, and then choose **Change Shutdown Behavior**.
   c. If the **Shutdown behavior** is set to **Terminate**, choose **Stop**, and then choose **Apply**.

      If the **Shutdown behavior** is already set to **Stop**, then choose **Cancel**.

5. Stop the instance. For more information about how to stop an instance, see Stopping and Starting Your Instances (p. 220).

   Warning
   When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

6. Create a snapshot of the volume to expand.

   a. In the navigation pane, choose **Volumes**, and then locate the volume you want to expand.
   b. From the context-menu (right-click) choose the volume that you want to expand, and then choose **Create Snapshot**.
   c. Enter information in the **Name** and **Description** fields, and then choose **Yes, Create**.

7. Create a new volume from the snapshot.

   a. In the navigation pane, chose **Snapshots**.
   b. When the status of the snapshot that you just created is set to **completed**, choose the snapshot, and then from the context-menu (right-click) choose **Create Volume**.
   c. In the **Create Volume** dialog box, choose the desired volume type and enter the new volume size. You must also set the **Availability Zone** to match the instance Availability Zone. Choose **Yes, Create**.

      Important
      If you do not set the **Availability Zone** to match the instance then you will not be able to attach the new volume to the instance.

8. Detach the old volume.

   a. In the navigation pane, choose **Volumes**, and then choose the old volume from the list. Make a note of the device name in the **Attachment Information** column:

      ```
      i-xxxxxxxx (instance_name):device_name
      ```

   b. From the context-menu (right-click) choose the old volume, and then choose **Detach Volume**.
   c. In the **Detach Volume** dialog box, choose **Yes, Detach**. It may take several minutes for the volume to detach.

9. Attach the newly expanded volume

a. In the navigation pane, choose **Volumes**.
b. From the context-menu (right-click) choose the new volume, and then choose **Attach Volume**.
c. Start typing the name or ID of the instance in the **Instance** field, and then choose the instance.
d. Enter the device name, for example /dev/sda1 (for a root volume), and then choose **Yes, Attach**.

10. Restart the instance.

    a. In the navigation pane, choose **Instances** and then choose the instance you want to restart.
    b. From the context-menu (right-click) choose **Instance State**, and then choose **Start**.
    c. In the **Start Instances** dialog box, choose **Yes, Start**. If the instance fails to start, and the volume being expanded is a root volume, verify that you attached the expanded volume using the same device name as the original volume, for example /dev/sda1.

       Important
       Only instances running in a VPC retain their public and Elastic IP addresses when they are stopped. If your instance is running in EC2-Classic, the EIP address is disassociated when the instance is stopped, so you must re-associate the EIP after restarting the instance. For more information, see Elastic IP Addresses (p. 485). If your instance is running in EC2-Classic but is not using an EIP, you must retrieve the new public DNS name for your instance in order to connect to it after it restarts.

After the instance has started, you can check the file system size to see if your instance recognizes the larger volume space.

If the size does not reflect your newly-expanded volume, you must extend the file system your device so that your instance can use the new space. For more information, see Extending a Windows File System (p. 546).

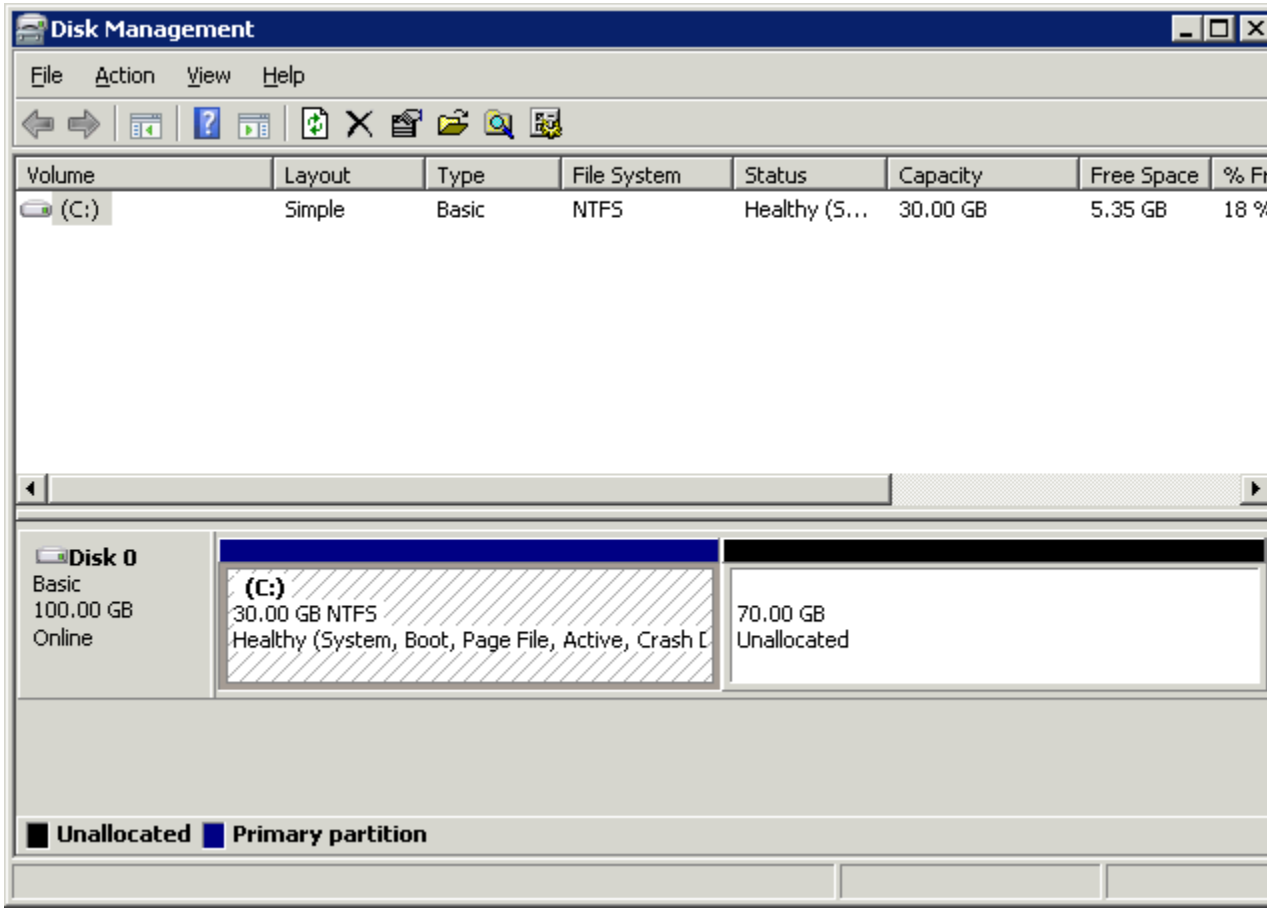
Extending a Windows File System

In Windows, you use the Disk Management utility to extend the disk size to the new size of the volume.

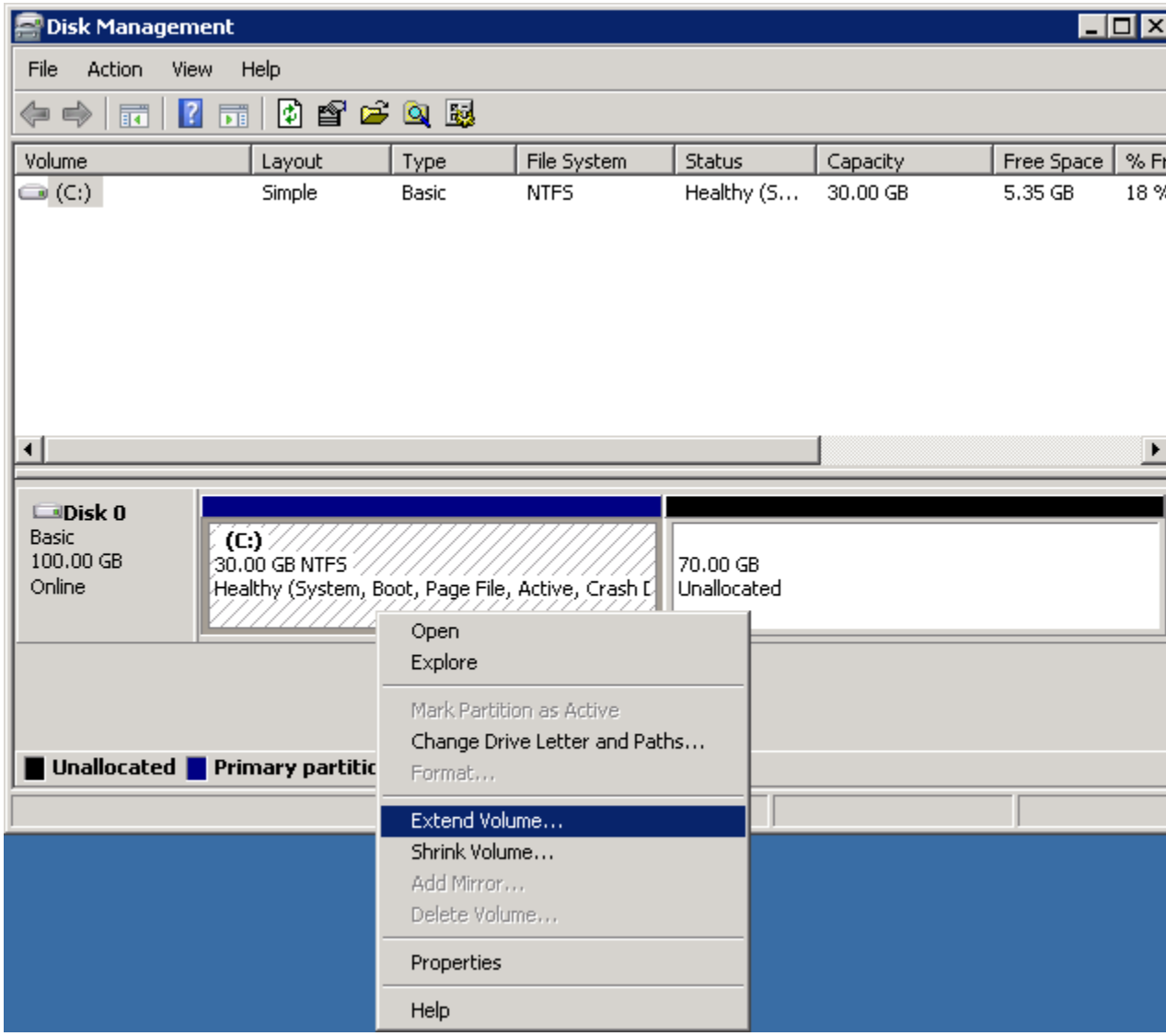
To extend a Windows file system

1. Log in to your Windows instance using Remote Desktop.
2. Open the Disk Management utility.

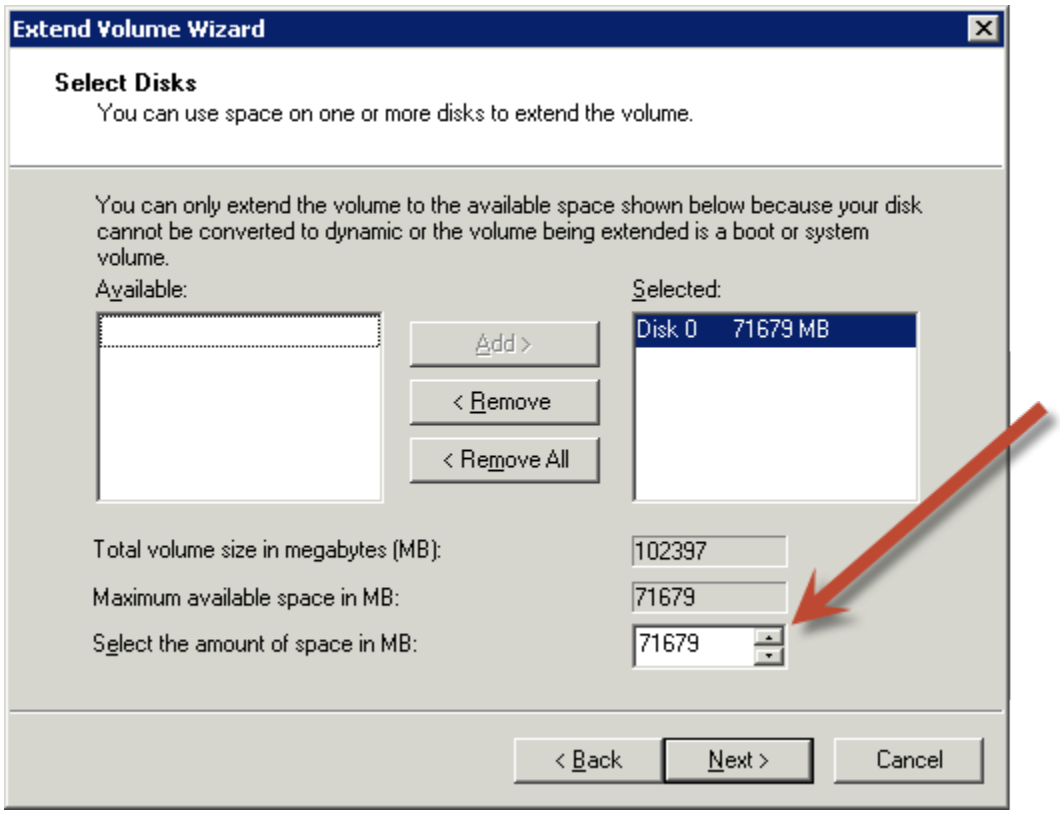
   - On the Windows Server 2012 Start screen, type **disk management** and choose **Create and format hard disk partitions** in the Search pane.
   - On Windows Server 2008 type **diskmgmt.msc** in the **Run** dialog and press **Enter**.



3. Right-click the expanded drive and select **Extend Volume**.



4. In the Extend Volume Wizard, choose **Next**, then set the **Select the amount of space in MB** field to the number of megabytes by which to extend the volume. Normally, you set this to the maximum available space. Complete the wizard.



Deleting the Old Volume

After the new volume has been attached and extended in the instance, you can delete the old volume if it is no longer needed.

To delete the old volume

1. In the Amazon EC2 console, choose **Volumes** in the navigation pane and then choose the volume you want to delete.
2. From the context-menu (right-click) choose **Delete Volume**.
3. In the **Delete Volume** dialog box, choose **Yes, Delete**.

Amazon EBS Snapshots

You can back up the data on your EBS volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. When you delete a snapshot, only the data exclusive to that snapshot is removed. Active snapshots contain all of the information needed to restore your data (from the time the snapshot was taken) to a new EBS volume.

If you are dealing with snapshots of sensitive data, you should consider encrypting your data manually before taking the snapshot or storing the data on a volume that is enabled with Amazon EBS encryption. For more information, see Amazon EBS Encryption (p. 558).

Contents

- Snapshot Overview (p. 550)

- Creating an Amazon EBS Snapshot (p. 550)
- Deleting an Amazon EBS Snapshot (p. 551)
- Copying an Amazon EBS Snapshot (p. 552)
- Viewing Amazon EBS Snapshot Information (p. 554)
- Sharing an Amazon EBS Snapshot (p. 554)

Snapshot Overview

When you create an EBS volume, you can create it based on an existing snapshot. The new volume begins as an exact replica of the original volume that was used to create the snapshot. When you create a volume from an existing snapshot, it loads lazily in the background so that you can begin using them right away. If you access a piece of data that hasn't been loaded yet, the volume immediately downloads the requested data from Amazon S3, and then continues loading the rest of the volume's data in the background. For more information, see Creating an Amazon EBS Snapshot (p. 550).

Snapshots of encrypted volumes are automatically encrypted. Volumes that are created from encrypted snapshots are also automatically encrypted. Your encrypted volumes and any associated snapshots always remain protected. For more information, see Amazon EBS Encryption (p. 558).

You can share your unencrypted snapshots with specific AWS accounts, make them public to share them with the entire AWS community. Users with access to your snapshots can create their own EBS volumes from your snapshot. This doesn't affect your snapshot. For more information about how to share snapshots, see Sharing an Amazon EBS Snapshot (p. 554).

Snapshots are constrained to the region in which they are created. After you have created a snapshot of an EBS volume, you can use it to create new volumes in the same region. For more information, see Restoring an Amazon EBS Volume from a Snapshot (p. 527). You can also copy snapshots across regions, making it easier to leverage multiple regions for geographical expansion, data center migration, and disaster recovery. You can copy any accessible snapshots that have a `completed` status. For more information, see Copying an Amazon EBS Snapshot (p. 552).

Creating an Amazon EBS Snapshot

After writing data to an EBS volume, you can periodically create a snapshot of the volume to use as a baseline for new volumes or for data backup. If you make periodic snapshots of a volume, the snapshots are incremental so that only the blocks on the device that have changed after your last snapshot are saved in the new snapshot. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.

Snapshots occur asynchronously; the point-in-time snapshot is created immediately, but the status of the snapshot is `pending` until the snapshot is complete (when all of the modified blocks have been transferred Amazon S3), which can take several hours for large initial snapshots or subsequent snapshots where many blocks have changed.

> **Important**
> Although you can take a snapshot of a volume while a previous snapshot of that volume is in the `pending` status, having multiple `pending` snapshots of a volume may result in reduced volume performance until the snapshots complete.
> There is a limit of 5 `pending` snapshots for a single volume. If you receive a `ConcurrentSnapshotLimitExceeded` error while trying to create multiple concurrent snapshots of the same volume, wait for one or more of the `pending` snapshots to complete before creating another snapshot of that volume.

Snapshots that are taken from encrypted volumes are automatically encrypted. Volumes that are created from encrypted snapshots are also automatically encrypted. Your encrypted volumes and any associated snapshots always remain protected. For more information, see Amazon EBS Encryption (p. 558).

By default, only you can create volumes from snapshots that you own. However, you can choose to share your unencrypted snapshots with specific AWS accounts or make them public. For more information, see Sharing an Amazon EBS Snapshot (p. 554).

When a snapshot is created from a volume with an AWS Marketplace product code, the product code is propagated to the snapshot.

You can take a snapshot of an attached volume that is in use. However, snapshots only capture data that has been written to your Amazon EBS volume at the time the snapshot command is issued. This might exclude any data that has been cached by any applications or the operating system. If you can pause any file writes to the volume long enough to take a snapshot, your snapshot should be complete. However, if you can't pause all file writes to the volume, you should unmount the volume from within the instance, issue the snapshot command, and then remount the volume to ensure a consistent and complete snapshot. You can remount and use your volume while the snapshot status is `pending`.

To create a snapshot for Amazon EBS volumes that serve as root devices, you should stop the instance before taking the snapshot.

To unmount the volume in Windows, open Disk Management, right-click the volume to unmount, and select **Change Drive Letter and Path**. Select the mount point to remove, and then click **Remove**.

To create a snapshot using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Snapshots** in the navigation pane.
3. Choose **Create Snapshot**.
4. In the **Create Snapshot** dialog box, select the volume to create a snapshot for, and then choose **Create**.

To create a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- create-snapshot (AWS CLI)
- ec2-create-snapshot (Amazon EC2 CLI)
- New-EC2Snapshot (AWS Tools for Windows PowerShell)

Deleting an Amazon EBS Snapshot

When you delete a snapshot, only the data exclusive to that snapshot is removed. Deleting previous snapshots of a volume do not affect your ability to restore volumes from later snapshots of that volume.

If you make periodic snapshots of a volume, the snapshots are incremental so that only the blocks on the device that have changed since your last snapshot are saved in the new snapshot. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.

Note that you can't delete a snapshot of the root device of an EBS volume used by a registered AMI. You must first deregister the AMI before you can delete the snapshot. For more information, see Deregistering Your AMI (p. 76).

To delete a snapshot using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Snapshots** in the navigation pane.
3. Select a snapshot and then choose **Delete** from the **Actions** list.
4. Choose **Yes, Delete**.

To delete a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- delete-snapshot (AWS CLI)
- ec2-delete-snapshot (Amazon EC2 CLI)
- Remove-EC2Snapshot (AWS Tools for Windows PowerShell)

Copying an Amazon EBS Snapshot

With Amazon EBS, you can create point-in-time snapshots of volumes which we store for you in Amazon Simple Storage Service (Amazon S3). After you've created a snapshot and it has finished copying to Amazon S3 (when the snapshot status is `completed`), you can copy it from one AWS region to another, or within the same region. Snapshots are copied with Amazon S3 server-side encryption (256-bit Advanced Encryption Standard) to encrypt your data and the snapshot copy receives a snapshot ID that's different from the original snapshot's ID.

> **Note**
> To copy an Amazon Relational Database Service (Amazon RDS) snapshot, see Copying a DB Snapshot in the Amazon Relational Database Service User Guide.

You can use a copy of an snapshot in the following ways:

- Geographic expansion: Launch your applications in a new region.
- Migration: Move an application to a new region, to enable better availability and minimize cost.
- Disaster recovery: Back up your data and logs across different geographical locations at regular intervals. In case of disaster, you can restore your applications using point-in-time backups stored in the secondary region. This minimizes data loss and recovery time.

User-defined tags are not copied from the source snapshot to the new snapshot. After the copy operation is complete, you can apply user-defined tags to the new snapshot. For more information, see Tagging Your Amazon EC2 Resources (p. 609).

You can have up to five snapshot copy requests in progress to a single destination per account. You can copy any accessible snapshots that have a `completed` status, including shared snapshots and snapshots that you've created. You can also copy AWS Marketplace, VM Import/Export, and AWS Storage Gateway snapshots, but you must verify that the snapshot is supported in the destination region.

When you copy a snapshot, you are only charged for the data transfer and storage used to copy the snapshot data across regions and to store the copied snapshot in the destination region. You are not charged if the snapshot copy fails. However, if you cancel a snapshot copy that is not yet complete, or delete the source snapshot while the copy is in progress, you are charged for the bandwidth of the data transferred.

The first snapshot copy to another region is always a full copy. Each subsequent snapshot copy is incremental (which makes the copy process faster), meaning that only the blocks in the snapshot that have changed after your last snapshot copy to the same destination are transferred. Support for incremental

snapshots is specific to a region pair where a previous complete snapshot copy of the source volume is already available in the destination region, and it is limited to the default EBS CMK for encrypted snapshots. For example, if you copy an unencrypted snapshot from the US East (N. Virginia) region to the US West (Oregon) region, the first snapshot copy of the volume is a full copy and subsequent snapshot copies of the same volume transferred between the same regions are incremental.

Note
Snapshot copies within a single region do not copy any data at all as long as the following conditions apply:

- The encryption status of the snapshot copy does not change during the copy operation
- For encrypted snapshots, both the source snapshot and the copy are encrypted with the default EBS CMK

If you would like another account to be able to copy your snapshot, you must either modify the snapshot permissions to allow access to that account or make the snapshot public so that all AWS accounts may copy it. For more information, see Sharing an Amazon EBS Snapshot (p. 554).

Encrypted Snapshots

When you copy a snapshot, you can choose to encrypt the copy (if the original snapshot was not encrypted) or you can specify a different CMK than the original, and the resulting copied snapshot will use the new CMK. However, changing the encryption status of a snapshot or using a non-default EBS CMK during a copy operation always results in a full copy (not incremental), which may incur greater data transfer and storage charges. Encrypted snapshots cannot be shared between accounts or made public.

To copy a snapshot using the Amazon EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Snapshots**.
3. Select the snapshot to copy, and then choose **Copy** from the **Actions** list.
4. In the **Copy Snapshot** dialog box, update the following as necessary:
   - **Destination region**: Select the region where you want to write the copy of the snapshot.
   - **Description**: By default, the description includes information about the source snapshot so that you can identify a copy from the original. You can change this description as necessary.
   - **Encryption**: If the source snapshot is not encrypted, you can choose to encrypt the copy. You cannot decrypt an encrypted snapshot.
   - **Master Key**: The customer master key (CMK) that will be used to encrypt this snapshot. You can select from master keys in your account or type/paste the ARN of a key from a different account. You can create a new master encryption key in the IAM console.
5. Choose **Copy**.
6. In the **Copy Snapshot** confirmation dialog box, choose **Snapshots** to go to the **Snapshots** page in the region specified, or choose **Close**.

   To view the progress of the copy process later, switch to the destination region, and then refresh the **Snapshots** page. Copies in progress are listed at the top of the page.

To copy a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- copy-snapshot (AWS CLI)

- ec2-copy-snapshot (Amazon EC2 CLI)
- Copy-EC2Snapshot (AWS Tools for Windows PowerShell)

Viewing Amazon EBS Snapshot Information

You can view detailed information about your snapshots.

To view snapshot information using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Snapshots** in the navigation pane.
3. To reduce the list, choose an option from the **Filter** list. For example, to view only your snapshots, choose **Owned By Me**. You can filter your snapshots further by using the advanced search options. Choose the search bar to view the filters available.
4. To view more information about a snapshot, choose it.

To view snapshot information using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-snapshots (AWS CLI)
- ec2-describe-snapshots (Amazon EC2 CLI)
- Get-EC2Snapshot (AWS Tools for Windows PowerShell)

Sharing an Amazon EBS Snapshot

You can share your unencrypted snapshots with your co-workers or others in the AWS community by modifying the permissions of the snapshot. Users that you have authorized can quickly use your unencrypted shared snapshots as the basis for creating their own EBS volumes. If you choose, you can also make your unencrypted snapshots available publicly to all AWS users.

Users to whom you have granted access can copy your snapshot or create their own EBS volumes based on your snapshot, and your original snapshot remains intact.

Snapshots are constrained to the region in which they are created. If you would like to share a snapshot with another region, you need to copy the snapshot to that region. For more information about copying snapshots, see Copying an Amazon EBS Snapshot (p. 552).

Making your snapshot public shares all snapshot data with everyone; however, snapshots with AWS Marketplace product codes cannot be made public. Encrypted snapshots cannot be shared between accounts or made public.

> **Important**
> When you share a snapshot (whether by sharing it with another AWS account or making it public to all), you are giving others access to all the data on your snapshot. Share snapshots only with people with whom you want to share *all* your snapshot data.

To modify snapshot permissions using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Snapshots** in the navigation pane.
3. Select a snapshot and then choose **Modify Snapshot Permissions** from the **Actions** list.
4. Choose whether to make the snapshot public or to share it with specific AWS accounts:

- To make the snapshot public, choose **Public** (this is not a valid option for snapshots with AWS Marketplace product codes).
- To expose the snapshot to only specific AWS accounts, choose **Private**, enter the ID of the AWS account (without hyphens) in the **AWS Account Number** field, and choose **Add Permission**. Repeat until you've added all the required AWS accounts.

5. Choose **Save**.

To view and modify snapshot permissions using the command line

To view the `createVolumePermission` attribute of a snapshot, you can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-snapshot-attribute (AWS CLI)
- ec2-describe-snapshot-attribute (Amazon EC2 CLI)
- Get-EC2SnapshotAttribute (AWS Tools for Windows PowerShell)

To modify the `createVolumePermission` attribute of a snapshot, you can use one of the following commands.

- modify-snapshot-attribute (AWS CLI)
- ec2-modify-snapshot-attribute (Amazon EC2 CLI)
- Edit-EC2SnapshotAttribute (AWS Tools for Windows PowerShell)

Amazon EBS–Optimized Instances

An Amazon EBS–optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. This optimization provides the best performance for your EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.

EBS–optimized instances deliver dedicated throughput to Amazon EBS, with options between 500 Mbps and 4,000 Mbps, depending on the instance type you use. When attached to an EBS–optimized instance, General Purpose (SSD) volumes are designed to deliver within 10 percent of their baseline and burst performance 99.9 percent of the time in a given year, and Provisioned IOPS (SSD) volumes are designed to deliver within 10 percent of their provisioned performance 99.9 percent of the time in a given year. For more information, see Amazon EBS Volume Types (p. 520).

When you enable EBS optimization for an instance that is not EBS–optimized by default, you pay an additional low, hourly fee for the dedicated capacity. For pricing information, see EBS-optimized Instances on the Amazon EC2 Pricing page.

Contents

- Instance Types that Support EBS Optimization (p. 555)
- Enabling EBS Optimization at Launch (p. 557)
- Modifying EBS Optimization for a Running Instance (p. 557)

Instance Types that Support EBS Optimization

The following table shows which instance types support EBS optimization, the dedicated throughput to Amazon EBS, the maximum amount of IOPS the instance can support if you are using a 16 KB I/O size, and the approximate maximum bandwidth available on that connection in MB/s. Choose an EBS–optimized

instance that provides more dedicated EBS throughput than your application needs; otherwise, the connection between Amazon EBS and Amazon EC2 can become a performance bottleneck.

Note that some instance types are EBS–optimized by default. For instances that are EBS–optimized by default, there is no need to enable EBS optimization and there is no effect if you disable EBS optimization using the CLI or API. You can enable EBS optimization for the other instance types that support EBS optimization when you launch the instances, or enable EBS optimization after the instances are running.

Instance type	EBS-optimized by default	Throughput (Mbps)*	Max 16K IOPS**	Max bandwidth (MB/s)**
c1.xlarge		1,000	8,000	125
c3.xlarge		500	4,000	62.5
c3.2xlarge		1,000	8,000	125
c3.4xlarge		2,000	16,000	250
c4.large	Yes	500	4,000	62.5
c4.xlarge	Yes	750	6,000	93.75
c4.2xlarge	Yes	1,000	8,000	125
c4.4xlarge	Yes	2,000	16,000	250
c4.8xlarge	Yes	4,000	32,000	500
d2.xlarge	Yes	750	6,000	93.75
d2.2xlarge	Yes	1,000	8,000	125
d2.4xlarge	Yes	2,000	16,000	250
d2.8xlarge	Yes	4,000	32,000	500
g2.2xlarge		1,000	8,000	125
i2.xlarge		500	4,000	62.5
i2.2xlarge		1,000	8,000	125
i2.4xlarge		2,000	16,000	250
m1.large		500	4,000	62.5
m1.xlarge		1,000	8,000	125
m2.2xlarge		500	4,000	62.5
m2.4xlarge		1,000	8,000	125
m3.xlarge		500	4,000	62.5
m3.2xlarge		1,000	8,000	125
m4.large	Yes	450	3,600	56.25
m4.xlarge	Yes	750	6,000	93.75
m4.2xlarge	Yes	1,000	8,000	125
m4.4xlarge	Yes	2,000	16,000	250

Instance type	EBS-optimized by default	Throughput (Mbps)*	Max 16K IOPS**	Max bandwidth (MB/s)**
m4.10xlarge	Yes	4,000	32,000	500
r3.xlarge		500	4,000	62.5
r3.2xlarge		1,000	8,000	125
r3.4xlarge		2,000	16,000	250

Enabling EBS Optimization at Launch

You can enable EBS optimization for an instance by setting its EBS–optimized attribute.

To enable EBS optimization when launching an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Click **Launch Instance**. In **Step 1: Choose an Amazon Machine Image (AMI)**, select an AMI.
3. In **Step 2: Choose an Instance Type**, select an instance type that is listed as supporting EBS optimization.
4. In **Step 3: Configure Instance Details**, complete the fields that you need and select **Launch as EBS-optimized instance**. If the instance type that you selected in the previous step doesn't support EBS optimization, this option is not present. If the instance type that you selected is EBS–optimized by default, this option is selected and you can't deselect it.
5. Follow the directions to complete the wizard and launch your instance.

To enable EBS optimization when launching an instance using the command line

You can use one of the following options with the corresponding command. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- `--ebs-optimized` with run-instances (AWS CLI)
- `--ebs-optimized` with ec2-run-instances (Amazon EC2 CLI)
- `-EbsOptimized` with New-EC2Instance (AWS Tools for Windows PowerShell)

Modifying EBS Optimization for a Running Instance

You can enable or disable EBS optimization for a running instance by modifying its EBS–optimized instance attribute.

To enable EBS optimization for a running instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Instances**, and select the instance.
3. Click **Actions**, select **Instance State**, and then click **Stop**.

   Warning
   When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.
4. In the confirmation dialog box, click **Yes, Stop**. It can take a few minutes for the instance to stop.

5. With the instance still selected, click **Actions**, select **Instance Settings**, and then click **Change Instance Type**.
6. In the **Change Instance Type** dialog box, do one of the following:
   - If the instance type of your instance is EBS–optimized by default, **EBS-optimized** is selected and you can't deselect it. You can click **Cancel**, because EBS optimization is already enabled for the instance.
   - If the instance type of your instance supports EBS optimization, select **EBS-optimized**, and then click **Apply**.
   - If the instance type of your instance does not support EBS optimization, **EBS-optimized** is deselected and you can't select it. You can select an instance type from **Instance Type** that supports EBS optimization, select **EBS-optimized**, and then click **Apply**.
7. Click **Actions**, select **Instance State**, and then click **Start**.

To enable EBS optimization for a running instance using the command line

You can use one of the following options with the corresponding command. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- `--ebs-optimized` with modify-instance-attribute (AWS CLI)
- `--ebs-optimized` with ec2-modify-instance-attribute (Amazon EC2 CLI)
- `-EbsOptimized` with Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

Amazon EBS Encryption

Amazon EBS encryption offers you a simple encryption solution for your EBS volumes without the need for you to build, maintain, and secure your own key management infrastructure. When you create an encrypted EBS volume and attach it to a supported instance type, data stored at rest on the volume, disk I/O, and snapshots created from the volume are all encrypted. The encryption occurs on the servers that host EC2 instances, providing encryption of data-in-transit from EC2 instances to EBS storage.

Amazon EBS encryption uses AWS Key Management Service (AWS KMS) customer master keys (CMK) when creating encrypted volumes and any snapshots created from your encrypted volumes. The first time you create an encrypted volume in a region, a default CMK is created for you automatically. This key is used for Amazon EBS encryption unless you select a CMK that you created separately using AWS KMS. Creating your own CMK gives you more flexibility, including the ability to create, rotate, disable, define access controls, and audit the encryption keys used to protect your data. For more information, see the AWS Key Management Service Developer Guide.

This feature is supported with all EBS volume types (General Purpose (SSD), Provisioned IOPS (SSD), and Magnetic), and you can expect the same IOPS performance on encrypted volumes as you would with unencrypted volumes, with a minimal effect on latency. You can access encrypted volumes the same way that you access existing volumes; encryption and decryption are handled transparently and they require no additional action from you, your EC2 instance, or your application.

Important
Encrypted boot volumes are not supported at this time.

The Amazon EBS encryption feature is also extended to snapshots of your encrypted volumes. Snapshots that are taken from encrypted volumes are automatically encrypted. Volumes that are created from encrypted snapshots are also automatically encrypted. Your encrypted volumes and any associated snapshots always remain protected.

Amazon EBS encryption is only available on select instance types. You can attach both encrypted and unencrypted volumes to a supported instance type. For more information, see Supported Instance Types (p. 559).

Contents

- Encryption Key Management (p. 559)
- Supported Instance Types (p. 559)
- Considerations (p. 560)
- Migrating Data (p. 560)

Encryption Key Management

Amazon EBS encryption handles key management for you. Each newly-created volume is encrypted with a unique 256-bit key; any snapshots of this volume and any subsequent volumes created from those snapshots also share that key. These keys are protected by our own key management infrastructure, which implements strong logical and physical security controls to prevent unauthorized access. Your data and associated keys are encrypted using the industry-standard AES-256 algorithm.

You cannot change the CMK that is associated with an existing snapshot or encrypted volume. However, you can associate a different CMK during a snapshot copy operation (including encrypting a copy of an unencrypted snapshot) and the resulting copied snapshot will use the new CMK.

Amazon's overall key management infrastructure uses Federal Information Processing Standards (FIPS) 140-2 approved cryptographic algorithms and is consistent with National Institute of Standards and Technology (NIST) 800-57 recommendations.

Each AWS account has a unique master key that is stored completely separate from your data, on a system that is surrounded with strong physical and logical security controls. Each encrypted volume (and its subsequent snapshots) is encrypted with a unique volume encryption key that is then encrypted with a region-specific secure master key. The volume encryption keys are used in memory on the server that hosts your EC2 instance; they are never stored on disk in plain text.

Supported Instance Types

Amazon EBS encryption is available on the instance types listed in the table below. These instance types leverage the Intel AES New Instructions (AES-NI) instruction set to provide faster and simpler data protection. You can attach both encrypted and unencrypted volumes to these instance types simultaneously.

Instance family	Instance types that support Amazon EBS encryption
General purpose	`m3.medium` | `m3.large` | `m3.xlarge` | `m3.2xlarge` | `m4.large` | `m4.xlarge` | `m4.2xlarge` | `m4.4xlarge` | `m4.10xlarge` | `t2.large`
Compute optimized	`c4.large` | `c4.xlarge` | `c4.2xlarge` | `c4.4xlarge` | `c4.8xlarge` | `c3.large` | `c3.xlarge` | `c3.2xlarge` | `c3.4xlarge` | `c3.8xlarge`
Memory optimized	`cr1.8xlarge` | `r3.large` | `r3.xlarge` | `r3.2xlarge` | `r3.4xlarge` | `r3.8xlarge`
Storage optimized	`d2.xlarge` | `d2.2xlarge` | `d2.4xlarge` | `d2.8xlarge` | `i2.xlarge` | `i2.2xlarge` | `i2.4xlarge` | `i2.8xlarge`
GPU instances	`g2.2xlarge` | `g2.8xlarge`

For more information about these instance types, see Instance Type Details.

Considerations

Snapshots that are taken from encrypted volumes are automatically encrypted with the same volume encryption key used to encrypt the volume. Volumes that are created from encrypted snapshots are also automatically encrypted with the same volume encryption key used to create the snapshot. There is no way to directly create an unencrypted volume from an encrypted snapshot; however, you can create an encrypted snapshot from an unencrypted snapshot by creating an encrypted copy of the unencrypted snapshot. For more information, see Copying an Amazon EBS Snapshot (p. 552).

Public or shared snapshots of encrypted volumes are not supported, because other accounts would be able to decrypt your data.

There is also no way to encrypt an existing volume. However, you can migrate existing data between encrypted volumes and unencrypted volumes. For more information, see To migrate data between encrypted and unencrypted volumes (p. 560).

Important
Encrypted boot volumes are not supported at this time.

Migrating Data

If you have existing data that you would like to store on an encrypted volume, you need to migrate the data from your unencrypted volume to a new encrypted volume.

Important
Encrypted boot volumes are not supported at this time.

Likewise, if you have data that currently resides on an encrypted volume that you would like to share with others, you need to migrate the data you want to share from your encrypted volume to a new unencrypted volume.

Note
To move data from an unencrypted volume to an encrypted volume, you can also create a snapshot of the unencrypted volume, create an encrypted copy of that snapshot, and then restore the encrypted snapshot to a new volume, which will also be encrypted.

To migrate data between encrypted and unencrypted volumes

1. Create your destination volume (encrypted or unencrypted, depending on your use case) by following the procedures in Creating an Amazon EBS Volume (p. 525).
2. Attach the destination volume to the instance that hosts the data you would like to migrate. For more information, see Attaching an Amazon EBS Volume to an Instance (p. 529).
3. Make the destination volume available by following the procedures in Making an Amazon EBS Volume Available for Use (p. 530). For Linux instances, you can create a mount point at `/mnt/destination` and mount the destination volume there.
4. Copy the data from your source directory to the destination volume.
   - **Linux**

     Use the **rsync** command as follows to copy the data from your source to the destination volume. In this example, the source data is located in `/mnt/source` and the destination volume is mounted at `/mnt/destination`.

     ```
     [ec2-user ~]$ sudo rsync -avh -E --progress /mnt/source/ /mnt/destination/
     ```

   - **Windows**

At a command prompt, use the **robocopy** command as follows to copy the data from your source to the destination volume. In this example, the source data is located in `D:\` and the destination volume is mounted at `E:\`.

```
PS C:\Users\Administrator> robocopy D:\ E:\ /e /copyall /eta
```

Amazon EBS Volume Performance on Windows Instances

Several factors can affect the performance of Amazon EBS volumes, such as instance configuration, I/O characteristics, workload demand, and storage configuration. After you learn the basics of working with EBS volumes, it's a good idea to look at the I/O performance you require and at your options for increasing EBS performance to meet those requirements.

For information about volume performance on Linux instances, see Amazon EBS Volume Performance on Linux Instances in the *Amazon EC2 User Guide for Linux Instances*.

Contents

- Amazon EBS Performance Tips (p. 561)
- Amazon EC2 Instance Configuration (p. 562)
- I/O Characteristics (p. 564)
- Workload Demand (p. 565)
- Pre-Warming Amazon EBS Volumes (p. 565)
- RAID Configuration on Windows (p. 567)
- Benchmark Volumes (p. 572)

Amazon EBS Performance Tips

- When you consider the performance requirements for your EBS storage application, it is important to start with an EC2 configuration that is optimized for EBS and that can handle the bandwidth that your application storage system requires. For more information, see Amazon EC2 Instance Configuration (p. 562).
- When you measure the performance of your EBS volumes, especially with General Purpose (SSD) and Provisioned IOPS (SSD) volumes, it is important to understand the units of measure involved and how performance is calculated. For more information, see I/O Characteristics (p. 564).
- There is a relationship between the maximum performance of your EBS volumes, the amount of I/O you are driving to them, and the amount of time it takes for each transaction to complete. Each of these factors (performance, I/O, and time) affects the others, and different applications are more sensitive to one factor or another. For more information, see Workload Demand (p. 565).
- There is a significant increase in latency when you first access each block of data on a newly created or restored EBS volume (General Purpose (SSD), Provisioned IOPS (SSD), or Magnetic). You can avoid this performance hit by accessing each block in advance. For more information, see Pre-Warming Amazon EBS Volumes (p. 565).
- Some instance types can drive more I/O throughput than what you can provision for a single Amazon EBS volume. You can join multiple General Purpose (SSD) or Provisioned IOPS (SSD) volumes together in a RAID 0 configuration to use the available bandwidth for these instances. You can also provide redundancy for your volumes with a RAID 1 (mirrored) configuration. For more information, see RAID Configuration on Windows (p. 567).

- You can benchmark your storage and compute configuration to make sure you achieve the level of performance you expect to see before taking your application live. For more information, see Benchmark Volumes (p. 572).
- Amazon Web Services provides performance metrics for EBS that you can analyze and view with Amazon CloudWatch and status checks that you can use to monitor the health of your volumes. For more information, see Monitoring the Status of Your Volumes (p. 532).
- Frequent snapshots provide a higher level of data durability, but they may slightly degrade the performance of your application while the snapshot is in progress. This trade off becomes critical when you have data that changes rapidly. Whenever possible, plan for snapshots to occur during off-peak times in order to minimize workload impact. For more information, see Amazon EBS Snapshots (p. 549).

Amazon EC2 Instance Configuration

When you plan and configure EBS volumes for your application, it is important to consider the configuration of the instances that you will attach the volumes to. In order to get the most performance out of your EBS volumes, you should attach them to an instance with enough bandwidth to support your volumes, such as an EBS-optimized instance or an instance with 10 Gigabit network connectivity. This is especially important when you use General Purpose (SSD) or Provisioned IOPS (SSD) volumes, or when you stripe multiple volumes together in a RAID configuration.

Use EBS-Optimized or 10 Gigabit Network Instances

Any performance-sensitive workloads that require minimal variability and dedicated Amazon EC2 to Amazon EBS traffic, such as production databases or business applications, should use General Purpose (SSD) or Provisioned IOPS (SSD) volumes that are attached to an EBS-optimized instance or an instance with 10 Gigabit network connectivity. EC2 instances that do not meet this criteria offer no guarantee of network resources. The only way to ensure sustained reliable network bandwidth between your EC2 instance and your EBS volumes is to launch the EC2 instance as EBS-optimized or choose an instance type with 10 Gigabit network connectivity. To see which instance types include 10 Gigabit network connectivity, see Instance Type Details.

Choose an EC2 Instance with Enough Bandwidth

Launching an instance that is EBS-optimized provides you with a dedicated connection between your EC2 instance and your EBS volume. However, it is still possible to provision EBS volumes that exceed the available bandwidth for certain instance types, especially when multiple volumes are striped in a RAID configuration. The following table shows which instance types are available to be launched as EBS-optimized, the dedicated throughput to Amazon EBS, the maximum amount of IOPS the instance can support if you are using a 16 KB I/O size, and the approximate I/O bandwidth available on that connection in MB/s. Be sure to choose an EBS-optimized instance that provides more dedicated EBS throughput than your application needs; otherwise, the Amazon EBS to Amazon EC2 connection will become a performance bottleneck.

Instance type	EBS-optimized by default	Throughput (Mbps)*	Max 16K IOPS**	Max bandwidth (MB/s)**
`c1.xlarge`		1,000	8,000	125
`c3.xlarge`		500	4,000	62.5
`c3.2xlarge`		1,000	8,000	125
`c3.4xlarge`		2,000	16,000	250
`c4.large`	Yes	500	4,000	62.5
`c4.xlarge`	Yes	750	6,000	93.75

Instance type	EBS-optimized by default	Throughput (Mbps)*	Max 16K IOPS**	Max bandwidth (MB/s)**
c4.2xlarge	Yes	1,000	8,000	125
c4.4xlarge	Yes	2,000	16,000	250
c4.8xlarge	Yes	4,000	32,000	500
d2.xlarge	Yes	750	6,000	93.75
d2.2xlarge	Yes	1,000	8,000	125
d2.4xlarge	Yes	2,000	16,000	250
d2.8xlarge	Yes	4,000	32,000	500
g2.2xlarge		1,000	8,000	125
i2.xlarge		500	4,000	62.5
i2.2xlarge		1,000	8,000	125
i2.4xlarge		2,000	16,000	250
m1.large		500	4,000	62.5
m1.xlarge		1,000	8,000	125
m2.2xlarge		500	4,000	62.5
m2.4xlarge		1,000	8,000	125
m3.xlarge		500	4,000	62.5
m3.2xlarge		1,000	8,000	125
m4.large	Yes	450	3,600	56.25
m4.xlarge	Yes	750	6,000	93.75
m4.2xlarge	Yes	1,000	8,000	125
m4.4xlarge	Yes	2,000	16,000	250
m4.10xlarge	Yes	4,000	32,000	500
r3.xlarge		500	4,000	62.5
r3.2xlarge		1,000	8,000	125
r3.4xlarge		2,000	16,000	250

The `m1.large` instance has a maximum 16 KB IOPS value of 4,000, but unless this instance type is launched as EBS-optimized, that value is an absolute best-case scenario and is not guaranteed; to consistently achieve 4,000 16 KB IOPS, you must launch this instance as EBS-optimized. However, if a 4,000 IOPS Provisioned IOPS (SSD) volume is attached to an EBS-optimized `m1.large` instance, the EC2 to EBS connection bandwidth limit prevents this volume from providing the 320 MB/s maximum aggregate throughput available to it. In this case, we must use an EBS-optimized EC2 instance that supports at least 320 MB/s of throughput, such as the `c4.8xlarge` instance type.

General Purpose (SSD) volumes have a throughput limit between 128 MB/s and 160 MB/s per volume (depending on volume size), which pairs well with a 1,000 Mbps EBS-optimized connection. Instance types that offer more than 1,000 Mbps of throughput to Amazon EBS can use more than one General Purpose (SSD) volume to take advantage of the available throughput. Provisioned IOPS (SSD) volumes have a throughput limit range of 256 KiB for each IOPS provisioned, up to a maximum of 320 MiB/s (at 1,280 IOPS). For more information, see Amazon EBS Volume Types (p. 520).

Instance types with 10 Gigabit network connectivity support up to 800 MB/s of throughput and 48,000 16K IOPS for unencrypted Amazon EBS volumes and up to 25,000 16K IOPS for encrypted Amazon EBS volumes. Because the maximum provisioned IOPS value for EBS volumes is 20,000 for Provisioned IOPS (SSD) volumes and 10,000 for General Purpose (SSD) volumes, you can use several EBS volumes simultaneously to reach the level of I/O performance available to these instance types. For more information about which instance types include 10 Gigabit network connectivity, see Instance Type Details.

You should use EBS-optimized instances when available to get the full performance benefits of Amazon EBS General Purpose (SSD) and Provisioned IOPS (SSD) volumes. For more information, see Amazon EBS–Optimized Instances (p. 555).

I/O Characteristics

On a given volume configuration, certain I/O characteristics drive the performance behavior on the back end. General Purpose (SSD) and Provisioned IOPS (SSD) volumes deliver consistent performance whether an I/O operation is random or sequential, and also whether an I/O operation is to read or write data. I/O size, however, does make an impact on IOPS because of the way they are measured. In order to fully understand how General Purpose (SSD) and Provisioned IOPS (SSD) volumes will perform in your application, it is important to know what IOPS are and how they are measured.

What are IOPS?

IOPS are input/output operations per second. Amazon EBS measures each I/O operation per second (that is 256 KiB or smaller) as one IOPS. I/O operations that are larger than 256 KiB are counted in 256 KiB capacity units. For example, a single 1,024 KiB I/O operation would count as 4 IOPS; however, 1,024 I/O operations at 1 KiB each would count as 1,024 IOPS.

When you create a 3,000 IOPS volume, either a 3,000 IOPS Provisioned IOPS (SSD) volume or a 1,000 GiB General Purpose (SSD) volume, and attach it to an EBS-optimized instance that can provide the necessary bandwidth, you can transfer up to 3,000 chunks of data per second (provided that the I/O does not exceed the per volume throughput limit of the volume.

I/O size and volume throughput limits

If your I/O chunks are very large, you may experience a smaller number of IOPS than you provisioned because you are hitting the throughput limit of the volume. For example 1,000 GiB General Purpose (SSD) volume has an IOPS limit of 3,000 and a volume throughput limit of 160 MiB/s. If you are using a 256 KiB I/O size, your volume will reach its throughput limit at 640 IOPS (640 x 256 KiB = 160 MiB). For smaller I/O sizes (such as 16 KiB), this same volume can sustain 3,000 IOPS because the throughput is well below 160 MiB/s. For more information on the throughput limits for each EBS volume type, see Amazon EBS Volume Types (p. 520).

For smaller I/O operations, you may even see an IOPS value that is higher than what you have provisioned (when measured on the client side), and this is because the client operating system may be coalescing multiple smaller I/O operations into a smaller number of large chunks.

If you are not experiencing the expected IOPS or throughput you have provisioned, ensure that your EC2 bandwidth is not the limiting factor; your instance should be EBS-optimized (or include 10 Gigabit network connectivity) and your instance type EBS dedicated bandwidth should exceed the I/O throughput you intend to drive. For more information, see Amazon EC2 Instance Configuration (p. 562). Another possible cause for not experiencing the expected IOPS is that you are not driving enough I/O to the EBS volumes. For more information, see Workload Demand (p. 565).

Workload Demand

Workload demand plays an important role in getting the most out of your General Purpose (SSD) and Provisioned IOPS (SSD) volumes. In order for your volumes to deliver the amount of IOPS that are available, they need to have enough I/O requests sent to them. There is a relationship between the demand on the volumes, the amount of IOPS that are available to them, and the latency of the request (the amount of time it takes for the I/O operation to complete).

Average Queue Length

The queue length is the number of pending I/O requests for a device. Optimal average queue length will vary for every customer workload, and this value depends on your particular application's sensitivity to IOPS and latency. If your workload is not delivering enough I/O requests to maintain your optimal average queue length, then your volume might not consistently deliver the IOPS that you have provisioned. However, if your workload maintains an average queue length that is higher than your optimal value, then your per-request I/O latency will increase; in this case, you should provision more IOPS for your volume.

To determine the optimal average queue length for your workload, we recommend that you target a queue length of 1 for every 500 IOPS available (baseline for General Purpose (SSD) volumes and the provisioned amount for Provisioned IOPS (SSD) volumes). Then you can monitor your application performance and tune that value based on your application requirements. For example, a volume with 1,000 provisioned IOPS should target an average queue length of 2 and tune that value up or down to see what performs best for your application.

> **Note**
> Per-request I/O latency may increase with higher average queue lengths.

Latency

Latency is the true end-to-end client time of an I/O operation; in other words, when the client sends a IO, how long does it take to get an acknowledgement from the storage subsystem that the IO read or write is complete. If your I/O latency is higher than you require, check your average queue length to make sure that your application is not trying to drive more IOPS than you have provisioned. You can maintain high IOPS while keeping latency down by maintaining a low average queue length. Consistently driving a greater number of IOPS to a volume than it has available to it (or provisioned) can also cause increased latency times; if your application requires a greater number of IOPS than your volume can provide, you should consider using a larger General Purpose (SSD) volume with a higher base performance level or a Provisioned IOPS (SSD) volume with more provisioned IOPS to achieve faster latencies.

Pre-Warming Amazon EBS Volumes

When you create any new EBS volume (General Purpose (SSD), Provisioned IOPS (SSD), or Magnetic) or restore a volume from a snapshot, the back-end storage blocks are allocated to you immediately. However, the first time you access a block of storage, it must be either wiped clean (for new volumes) or instantiated from its snapshot (for restored volumes) before you can access the block. This preliminary action takes time and can cause a significant increase in the latency of an I/O operation the first time each block is accessed. For most applications, amortizing this cost over the lifetime of the volume is acceptable. Performance is restored after the data is accessed once.

However, you can avoid this performance hit in a production environment by writing to or reading from all of the blocks on your volume before you use it; this process is called *pre-warming*. Writing to all of the blocks on a volume is preferred, but that is not an option for volumes that were restored from a snapshot, because that would overwrite the restored data. For a completely new volume that was created from scratch, you should write to all blocks before using the volume. For a new volume created from a snapshot, you should read all the blocks that have data before using the volume.

> **Important**
> While pre-warming Provisioned IOPS (SSD) volumes that were restored from snapshots, the performance of the volume may drop below 50 percent of its expected level, which causes the

volume to display a `warning` state in the **I/O Performance** status check. This is expected, and you can ignore the `warning` state on Provisioned IOPS (SSD) volumes while you are pre-warming them. For more information, see Monitoring Volumes with Status Checks (p. 535).

Pre-Warming Amazon EBS Volumes on Windows

There are multiple ways to pre-warm EBS volumes on Windows. The most simple solution is to provide a full format of the volume. Use the following command to perform a full format of a new volume:

Warning
The following command will destroy any existing data on the volume.

```
C:\>format drive_letter: /p:1
```

You can also perform a full format by right-clicking on the drive in a Windows Explorer window and clicking **Format**. Because this operation destroys all data on the volume, it is only appropriate for *new* volumes. For a read-only pre-warming tool, which allows you to pre-warm volumes that have been restored from a snapshot or that contain existing data (such as the `C:` drive), you should consider **dd** for Windows.

To install dd for Windows

The **dd** for the Windows program provides a similar experience to the **dd** program that is commonly available for Linux and Unix systems, and it allows you to pre-warm Amazon EBS volumes that have been restored from snapshots. At the time of this writing, the most recent beta version contains the `/dev/null` virtual device that is required to pre-warm volumes restored from snapshots. Full documentation for the program is available at http://www.chrysocome.net/dd.

1. Download the most recent binary version of **dd** for Windows from http://www.chrysocome.net/dd. You must use version 0.6 beta 3 or newer to pre-warm restored volumes.
2. (Optional) Create a folder for command line utilities that is easy to locate and remember, such as `C:\bin`. If you already have a designated folder for command line utilities, you can use that folder instead in the following step.
3. Unzip the binary package and copy the `dd.exe` file to your command line utilities folder (for example, `C:\bin`).
4. Add the command line utilities folder to your `Path` environment variable so you can execute the programs in that folder from anywhere.

   Important
   The following steps don't update the environment variables in your current command prompt windows. The command prompt windows that you open after you complete these steps will contain the updates. This is why it's necessary for you to open a new command prompt window to verify that your environment is set up properly.

   a. Click **Start**, right-click **Computer**, and then click **Properties**.
   b. Click **Advanced system settings**.
   c. Click **Environment Variables**.
   d. Under **System Variables**, click the variable called **Path** and then click **Edit**.
   e. In **Variable value**, append a semicolon and the location of your command line utility folder (`;C:\bin\`) to the end of the existing value.
   f. Click **OK** to close the **Edit System Variable** window.

To Pre-Warm a Volume Using dd for Windows

1. Use the **wmic** command to list the available disks on your system.

```
C:\>wmic diskdrive get size,deviceid
DeviceID            Size
\\.\PHYSICALDRIVE2  80517265920
\\.\PHYSICALDRIVE1  80517265920
\\.\PHYSICALDRIVE0  128849011200
\\.\PHYSICALDRIVE3  107372805120
```

Identify the disk you want to pre-warm in the following steps. The `C:` drive is on `\\.\PHYSICALDRIVE0`. You can use the **diskmgmt.msc** utility to compare drive letters to disk drive numbers if you are not sure which drive number to use.

2. Execute the following command to read all blocks on the specified device (and send the output to the `/dev/null` virtual device). This command safely pre-warms your existing data and any restored snapshots of volumes that were fully pre-warmed.

```
C:\>dd if=\\.\PHYSICALDRIVEn of=/dev/null bs=1M --progress --size
```

3. When the operation completes, you are ready to use your new volume. For more information, see Making an Amazon EBS Volume Available for Use (p. 530).

RAID Configuration on Windows

With Amazon EBS, you can use any of the standard RAID configurations that you can use with a traditional bare metal server, as long as that particular RAID configuration is supported by the operating system for your instance. This is because all RAID is accomplished at the software level. For greater I/O performance than you can achieve with a single volume, RAID 0 can stripe multiple volumes together; for on-instance redundancy, RAID 1 can mirror two volumes together.

Amazon EBS volume data is replicated across multiple servers in an Availability Zone to prevent the loss of data from the failure of any single component. This replication makes Amazon EBS volumes ten times more reliable than typical commodity disk drives. For more information, see Amazon EBS Availability and Durability in the Amazon EBS product detail pages.

If you need to create a RAID array on a Linux instance, see RAID Configuration on Linux in the *Amazon EC2 User Guide for Linux Instances*.

Contents

- RAID Configuration Options (p. 567)
- Creating a RAID Array on Windows (p. 568)

RAID Configuration Options

The following table compares the common RAID 0 and RAID 1 options.

Configuration	Use	Advantages	Disadvantages
RAID 0	When I/O performance is more important than fault tolerance; for example, as in a heavily used database (where data replication is already set up separately).	I/O is distributed across the volumes in a stripe. If you add a volume, you get the straight addition of throughput.	Performance of the stripe is limited to the worst performing volume in the set. Loss of a single volume results in a complete data loss for the array.

Configuration	Use	Advantages	Disadvantages
RAID 1	When fault tolerance is more important than I/O performance; for example, as in a critical application.	Safer from the standpoint of data durability.	Does not provide a write performance improvement; requires more Amazon EC2 to Amazon EBS bandwidth than non-RAID configurations because the data is written to multiple volumes simultaneously.

Important
RAID 5 and RAID 6 are not recommended for Amazon EBS because the parity write operations of these RAID modes consume some of the IOPS available to your volumes. Depending on the configuration of your RAID array, these RAID modes provide 20-30% fewer usable IOPS than a RAID 0 configuration. Increased cost is a factor with these RAID modes as well; when using identical volume sizes and speeds, a 2-volume RAID 0 array can outperform a 4-volume RAID 6 array that costs twice as much.

Creating a RAID 0 array allows you to achieve a higher level of performance for a file system than you can provision on a single Amazon EBS volume. A RAID 1 array offers a "mirror" of your data for extra redundancy. Before you perform this procedure, you need to decide how large your RAID array should be and how many IOPS you want to provision.

The resulting size of a RAID 0 array is the sum of the sizes of the volumes within it, and the bandwidth is the sum of the available bandwidth of the volumes within it. The resulting size and bandwidth of a RAID 1 array is equal to the size and bandwidth of the volumes in the array. For example, two 500 GiB Amazon EBS volumes with 4,000 provisioned IOPS each will create a 1000 GiB RAID 0 array with an available bandwidth of 8,000 IOPS and 640 MB/s of throughput or a 500 GiB RAID 1 array with an available bandwidth of 4,000 IOPS and 320 MB/s of throughput.

This documentation provides basic RAID setup examples. For more information about RAID configuration, performance, and recovery, see the Linux RAID Wiki at https://raid.wiki.kernel.org/index.php/Linux_Raid.

Creating a RAID Array on Windows

Use the following procedure to create the RAID array. Note that you can get directions for Linux instances from Creating a RAID Array on Linux in the *Amazon EC2 User Guide for Linux Instances*.

To create a RAID array on Windows

1. Create the Amazon EBS volumes for your array. For more information, see Creating an Amazon EBS Volume (p. 525).

   Important
   Create volumes with identical size and IOPS performance values for your array. Make sure you do not create an array that exceeds the available bandwidth of your EC2 instance. For more information, see Amazon EC2 Instance Configuration (p. 562).

2. Attach the Amazon EBS volumes to the instance that you want to host the array. For more information, see Attaching an Amazon EBS Volume to an Instance (p. 529).
3. Connect to your Windows instance. For more information, see Connecting to Your Windows Instance Using RDP (p. 216).
4. Open a command prompt and type the **diskpart** command.

```
PS C:\Users\Administrator> diskpart

Microsoft DiskPart version 6.1.7601
Copyright (C) 1999-2008 Microsoft Corporation.
On computer: WIN-BM6QPPL51CO
```

5. At the DISKPART prompt, list the available disks with the following command.

```
DISKPART> list disk

  Disk ###  Status         Size     Free     Dyn  Gpt
  --------  -------------  -------  -------  ---  ---
  Disk 0    Online           30 GB      0 B
  Disk 1    Online            8 GB      0 B
  Disk 2    Online            8 GB      0 B
  Disk 3    Online            8 GB      0 B
  Disk 4    Online            8 GB      0 B
  Disk 5    Online          419 GB      0 B
  Disk 6    Online          419 GB      0 B
```

   Identify the disks you want to use in your array and take note of their disk numbers.

6. Each disk you want to use in your array must be an online dynamic disk that does not contain any existing volumes. Use the following steps to convert basic disks to dynamic disks and to delete any existing volumes.

   a. Select a disk you want to use in your array with the following command, substituting *n* with your disk number.

   ```
   DISKPART> select disk n

   Disk n is now the selected disk.
   ```

   b. If the selected disk is listed as Offline, bring it online by running the **online disk** command.

   c. If the selected disk does not have an asterisk in the Dyn column in the previous **list disk** command output, you need to convert it to a dynamic disk.

   ```
   DISKPART> convert dynamic
   ```

   Note
   If you receive an error that the disk is write protected, you can clear the read-only flag with the **ATTRIBUTE DISK CLEAR READONLY** command and then try the dynamic disk conversion again.

   d. Use the **detail disk** command to check for existing volumes on the selected disk.

   ```
   DISKPART> detail disk

   XENSRC PVDISK SCSI Disk Device
   Disk ID: 2D8BF659
   Type   : SCSI
   Status : Online
   Path   : 0
   Target : 1
   ```

```
LUN ID : 0
Location Path : PCIROOT(0)#PCI(0300)#SCSI(P00T01L00)
Current Read-only State : No
Read-only  : No
Boot Disk  : No
Pagefile Disk  : No
Hibernation File Disk  : No
Crashdump Disk  : No
Clustered Disk  : No

  Volume ###  Ltr  Label        Fs     Type        Size     Status
Info
  ----------  ---  -----------  -----  ----------  -------  ---------
--------
  Volume 2     D   NEW VOLUME   FAT32  Simple      8189 MB  Healthy
```

Note any volume numbers on the disk. In this example, the volume number is 2. If there are no volumes, you can skip the next step.

e. (Only required if volumes were identified in the previous step) Select and delete any existing volumes on the disk that you identified in the previous step.

Warning
This destroys any existing data on the volume.

i. Select the volume, substituting *n* with your volume number.

```
DISKPART> select volume n
Volume n is the selected volume.
```

ii. Delete the volume.

```
DISKPART> delete volume

DiskPart successfully deleted the volume.
```

iii. Repeat these substeps for each volume you need to delete on the selected disk.

f. Repeat Step 6 (p. 569) for each disk you want to use in your array.

7. Verify that the disks you want to use are now dynamic.

```
DISKPART> list disk

  Disk ###  Status         Size     Free     Dyn  Gpt
  --------  -------------  -------  -------  ---  ---
  Disk 0    Online           30 GB      0 B
  Disk 1    Online            8 GB      0 B   *
  Disk 2    Online            8 GB      0 B   *
  Disk 3    Online            8 GB      0 B   *
* Disk 4    Online            8 GB      0 B   *
  Disk 5    Online          419 GB      0 B
  Disk 6    Online          419 GB      0 B
```

8. Create your raid array. On Windows, a RAID 0 volume is referred to as a striped volume and a RAID 1 volume is referred to as a mirrored volume.

   (Striped volumes only) To create a striped volume array on disks 1 and 2, use the following command (note the `stripe` option to stripe the array):

```
DISKPART> create volume stripe disk=1,2

DiskPart successfully created the volume.
```

   (Mirrored volumes only) To create a mirrored volume array on disks 3 and 4, use the following command (note the `mirror` option to mirror the array):

```
DISKPART> create volume mirror disk=3,4

DiskPart successfully created the volume.
```

9. Verify your new volume.

```
DISKPART> list volume

  Volume ###  Ltr  Label        Fs     Type        Size     Status     Info
  ----------  ---  -----------  -----  ----------  -------  ---------  ---
-----
  Volume 0     C                NTFS   Partition     29 GB  Healthy    System
* Volume 1                       RAW    Mirror      8190 MB  Healthy
  Volume 2                       RAW    Stripe        15 GB  Healthy
  Volume 5     Z   Temporary S  NTFS   Partition    419 GB  Healthy
  Volume 6     Y   Temporary S  NTFS   Partition    419 GB  Healthy
```

   Note that for this example the `Type` column lists a `Mirror` volume and a `Stripe` volume.

10. Select and format your volume so that you can begin using it.

    a. Select the volume you want to format, substituting *n* with your volume number.

```
DISKPART> select volume n

Volume n is the selected volume.
```

    b. Format the volume.

       Note
       To perform a full format, omit the `quick` option.

```
DISKPART> format quick recommended label="My new volume"

  100 percent completed

DiskPart successfully formatted the volume.
```

    c. Assign an available drive letter to your volume.

```
DISKPART> assign letter f

DiskPart successfully assigned the drive letter or mount point.
```

Your new volume is now ready to use.

Benchmark Volumes

This section demonstrates how you can test the performance of Amazon EBS volumes by simulating workloads similar to those of a database application. The process is as follows:

1. Launch an EBS-optimized instance
2. Create new Amazon EBS volumes
3. Attach the volumes to your EBS-optimized instance
4. Create a RAID array from the volumes, then format and mount it
5. Install a tool to benchmark I/O performance
6. Benchmark the I/O performance of your volumes
7. Delete your volumes and terminate your instance so that you don't continue to incur charges

Set Up Your Instance

To get optimal performance from General Purpose (SSD) and Provisioned IOPS (SSD) volumes, we recommend that you use an EBS-optimized instance. EBS-optimized instances deliver dedicated throughput between Amazon EC2 and Amazon EBS, with options between 500 and 4,000 Mbps, depending on the instance type.

To create an EBS-optimized instance, select **Launch as an EBS-Optimized instance** when launching the instance using the EC2 console, or specify **--ebs-optimized** when using the command line. Be sure that you launch one of the instance types that supports this option. For the example tests in this topic, we recommend that you launch an `m1.xlarge` instance. For more information, see Amazon EBS–Optimized Instances (p. 555).

To create a General Purpose (SSD) volume, select **General Purpose (SSD)** when creating the volume using the EC2 console, or specify **--type gp2** when using the command line. To create a Provisioned IOPS (SSD) volume, select **Provisioned IOPS (SSD)** when creating the volume using the EC2 console, or specify **--type io1 --iops** *iops* when using the command line. For information about attaching these volumes to your instance, see Attaching an Amazon EBS Volume to an Instance (p. 529).

For the example tests, we recommend that you create a RAID array with 6 volumes, which offers a high level of performance. Because you are charged by the gigabytes used (and the number of provisioned IOPS for Provisioned IOPS (SSD) volumes), not the number of volumes, there is no additional cost for creating multiple, smaller volumes and using them to create a stripe set. If you're using Oracle ORION to benchmark your volumes, it can simulate striping the same way that Oracle ASM does, so we recommend that you let ORION do the striping. If you are using a different benchmarking tool, you'll need to stripe the volumes yourself.

For information about creating a striped volume on Windows, see Create a Striped Volume in Windows.

On Windows, a full format of the volume pre-warms it. Use the `format <drive letter> /p:1` command to write zeros to the entire disk.

Important
Unless you pre-warm the volume, you might see a significant increase in latency for each block of data on the volume the first time you access it.

Install Benchmark Tools

The following are among the possible tools you can install and use to benchmark the performance of Amazon EBS volumes.

Tool	Description
fio	For benchmarking I/O performance. (Note that fio has a dependency on libaio-devel.)
Oracle Orion Calibration Tool	For calibrating the I/O performance of storage systems to be used with Oracle databases.
SQLIO	For calibrating the I/O performance of storage systems to be used with Microsoft SQL Server. For information about how to improve the performance of your Microsoft SQL Server databases, see Optimizing Databases on the MSDN website.

Example Benchmarking Commands

These benchmarking tools support a wide variety of test parameters. You should use commands that approximate the workloads your volumes will support. These commands are intended as examples to help you get started.

Run the following commands on an EBS-optimized instance with attached Amazon EBS volumes that have been pre-warmed.

When you are finished testing your volumes, see these topics for help cleaning up: Deleting an Amazon EBS Volume (p. 543) and Terminate Your Instance (p. 224).

fio Commands

Run **fio** on the stripe set that you created.

The following command performs 16 KB random write operations.

```
C:\> fio --directory=/media/p_iops_vol0
--name fio_test_file --direct=1 --rw=randwrite --bs=16k --size=1G
--numjobs=16 --time_based --runtime=180 --group_reporting --norandommap
```

The following command performs 16 KB random read operations.

```
C:\> fio --directory=/media/p_iops_vol0
--name fio_test_file --direct=1 --rw=randread --bs=16k --size=1G
--numjobs=16 --time_based --runtime=180 --group_reporting --norandommap
```

For more information about interpreting the results, see this tutorial: Inspecting disk IO performance with fio.

Oracle ORION Commands

Run ORION on the Amazon EBS volumes, having it simulate Oracle ASM striping instead of providing it with a stripe set that uses Windows striping.

In the directory where you installed ORION, create a file, `piops_test.lun`, to specify the volumes for your stripe set. The following example file specifies six volumes to be striped.

```
\\.\D:
\\.\E:
\\.\F:
\\.\G:
\\.\H:
\\.\I:
```

The following command performs 16 KB random I/O operations (80 percent reads and 20 percent writes), simulating 64 KB RAID-0 stripes.

```
C:\> orion -run advanced -testname piops_test -size_small 16 -size_large 16
-type rand -simulate raid0 -stripe 64 -write 80 -matrix detailed -num_disks 6
```

After the command is finished, ORION generates output files with the results in the same directory. For more information about ORION, see its Documentation.

SQLIO Commands

Run SQLIO on the stripe set that you created.

Create a file, `param.txt`, to specify your striped set. The contents of this file should look something like this (here, `d:\` corresponds to the striped set, and the test uses 6 threads and a 10 GB file).

```
d:\bigtestfile.dat 6 0x0 10240
```

The following command performs 16 KB random data writes.

```
C:\> sqlio -kW -s600 -frandom -t8 -o8 -b16 -LS -BH -Fparam.txt
```

The following command performs 16 KB random data reads.

```
C:\> sqlio -kR -s600 -frandom -t8 -o8 -b16 -LS -BH -Fparam.txt
```

The results are displayed in the Command Prompt window. For more information about SQLIO, see the `readme.txt` file in your SQLIO installation directory.

Amazon EBS Commands

The following table summarizes the available commands for Amazon EBS and the corresponding API actions.

Command/Action	Description
attach-volume (AWS CLI) ec2-attach-volume (Amazon EC2 CLI) AttachVolume	Attaches the specified volume to a specified instance, exposing the volume using the specified device name. A volume can be attached to only a single instance at any time. The volume and instance must be in the same Availability Zone. The instance must be in the `running` or `stopped` state.
copy-snapshot (AWS CLI) ec2-copy-snapshot (Amazon EC2 CLI) CopySnapshot	Copies a point-in-time snapshot of an EBS volume and stores it in Amazon S3. You can copy the snapshot within the same region or from one region to another. You can use the snapshot to create new EBS volumes or AMIs.
create-snapshot (AWS CLI) ec2-create-snapshot (Amazon EC2 CLI) CreateSnapshot	Creates a snapshot of the volume you specify. After the snapshot is created, you can use it to create volumes that contain exactly the same data as the original volume.
create-volume (AWS CLI) ec2-create-volume (Amazon EC2 CLI) CreateVolume	Creates an EBS volume using the specified size and type, or based on a previously created snapshot.
delete-snapshot (AWS CLI) ec2-delete-snapshot (Amazon EC2 CLI) DeleteSnapshot	Deletes the specified snapshot. This command does not affect any current EBS volumes, regardless of whether they were used to create the snapshot or were derived from the snapshot.
delete-volume (AWS CLI) ec2-delete-volume (Amazon EC2 CLI) DeleteVolume	Deletes the specified volume. The command does not delete any snapshots that were created from the volume.
describe-snapshot-attribute (AWS CLI) ec2-describe-snapshot-attribute (Amazon EC2 CLI) DescribeSnapshotAttribute	Describes attributes for a snapshot.
describe-snapshots (AWS CLI) ec2-describe-snapshots (Amazon EC2 CLI) DescribeSnapshots	Describes the specified snapshot. Describes all snapshots, including their source volume, snapshot initiation time, progress (percentage complete), and status (`pending`, `completed`, and so on.).

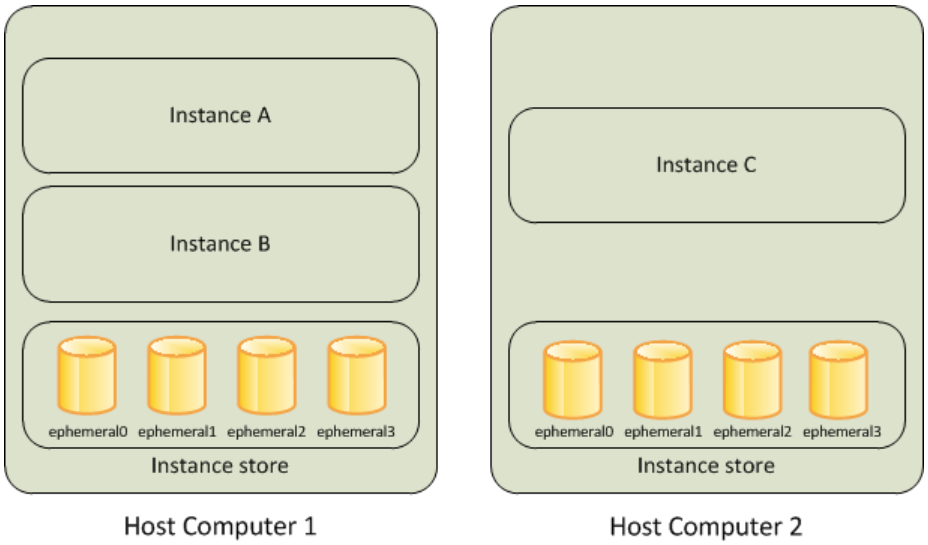
Command/Action	Description
describe-volume-attribute (AWS CLI) ec2-describe-volume-attribute (Amazon EC2 CLI) DescribeVolumeAttribute	Describes an attribute of a volume.
describe-volume-status (AWS CLI) ec2-describe-volume-status (Amazon EC2 CLI) DescribeVolumeStatus	Describes the status of one or more volumes. Volume status provides the result of the checks performed on your volumes to determine events that can impair the performance of your volumes.
describe-volumes (AWS CLI) ec2-describe-volumes (Amazon EC2 CLI) DescribeVolumes	Describes your volumes, including size, volume type, source snapshot, Availability Zone, creation time, status (`available` or `in-use`). If the volume is `in-use`, an attachment line shows the volume ID, the instance ID to which the volume is attached, the device name exposed to the instance, its status (attaching, attached, detaching, detached), and when it attached.
detach-volume (AWS CLI) ec2-detach-volume (Amazon EC2 CLI) DetachVolume	Detaches the specified volume from the instance it's attached to. This command does not delete the volume. The volume can be attached to another instance and will have the same data as when it was detached.
enable-volume-io (AWS CLI) ec2-enable-volume-io (Amazon EC2 CLI) EnableVolumeIO	Enables I/O operations for a volume that had I/O operations disabled because the data on the volume was potentially inconsistent.
modify-snapshot-attribute (AWS CLI) ec2-modify-snapshot-attribute (Amazon EC2 CLI) ModifySnapshotAttribute	Modifies permissions for a snapshot (i.e., who can create volumes from the snapshot). You can specify one or more AWS accounts, or specify `all` to make the snapshot public.
modify-volume-attribute (AWS CLI) ec2-modify-volume-attribute (Amazon EC2 CLI) ModifyVolumeAttribute	Modifies a volume's attributes to determine whether a volume should be automatically enabled for I/O operations.

Command/Action	Description
reset-snapshot-attribute (AWS CLI) ec2-reset-snapshot-attribute (Amazon EC2 CLI) ResetSnapshotAttribute	Resets permission settings for the specified snapshot.

Amazon EC2 Instance Store

An *instance store* provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. Instance store is ideal for temporary storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers.

An instance store consists of one or more instance store volumes exposed as block devices. The size of an instance store varies by instance type. The virtual devices for instance store volumes are `ephemeral[0-23]`. Instance types that support one instance store volume have `ephemeral0`. Instance types that support two instance store volumes have `ephemeral0` and `ephemeral1`, and so on. While an instance store is dedicated to a particular instance, the disk subsystem is shared among instances on a host computer.



Contents

- Instance Store Lifetime (p. 577)
- Instance Store Volumes (p. 578)
- Add Instance Store Volumes to Your EC2 Instance (p. 579)
- SSD Instance Store Volumes (p. 582)

Instance Store Lifetime

You can specify instance store volumes for an instance only when you launch it. The data in an instance store persists only during the lifetime of its associated instance. If an instance reboots (intentionally or

unintentionally), data in the instance store persists. However, data in the instance store is lost under the following circumstances:

- The underlying disk drive fails
- The instance stops
- The instance terminates

Therefore, do not rely on instance store for valuable, long-term data. Instead, you can build a degree of redundancy (for example, RAID 1/5/6), or use a file system (for example, HDFS and MapR-FS) that supports redundancy and fault tolerance. You can also back up data periodically to more durable data storage solutions such as Amazon S3 or Amazon EBS.

You can't detach an instance store volume from one instance and attach it to a different instance. If you create an AMI from an instance, the data on its instance store volumes isn't preserved and isn't present on the instance store volumes of the instances that you launch from the AMI.

Instance Store Volumes

The instance type of an instance determines the size of the instance store available for the instance, and the type of hardware used for the instance store volumes. Instance store volumes are included as part of the instance's hourly cost. You must specify the instance store volumes that you'd like to use when you launch the instance, and then format and mount them before using them. You can't make an instance store volume available after you launch the instance. For more information, see Add Instance Store Volumes to Your EC2 Instance (p. 579).

The instance type of an instance also determines the type of hardware for the instance store volumes. Some instance types use solid state drives (SSD) to deliver very high random I/O performance. This is a good option when you need storage with very low latency, but you don't need the data to persist when the instance terminates or you can take advantage of fault tolerant architectures. For more information see SSD Instance Store Volumes (p. 582).

The following table shows the size and quantity of the instance store volumes available to each instance type.

Instance Type	Instance Store Volumes
`c1.medium`	1 x 350 GB
`c1.xlarge`	4 x 420 GB (1680 GB)
`c3.large`	2 x 16 GB SSD (32 GB)
`c3.xlarge`	2 x 40 GB SSD (80 GB)
`c3.2xlarge`	2 x 80 GB SSD (160 GB)
`c3.4xlarge`	2 x 160 GB SSD (320 GB)
`c3.8xlarge`	2 x 320 GB SSD (640 GB)
`cc2.8xlarge`	4 x 840 GB (3360 GB)
`cg1.4xlarge`	2 x 840 GB (1680 GB)
`cr1.8xlarge`	2 x 120 GB SSD (240 GB)
`d2.xlarge`	3 x 2000 GB (6 TB)
`d2.2xlarge`	6 x 2000 GB (12 TB)

Instance Type	Instance Store Volumes
d2.4xlarge	12 x 2000 GB (24 TB)
d2.8xlarge	24 x 2000 GB (48 TB)
g2.2xlarge	1 x 60 GB SSD
g2.8xlarge	2 x 120 GB SSD
hi1.4xlarge	2 x 1024 GB SSD (2048 GB)
hs1.8xlarge	24 x 2000 GB (48 TB)
i2.xlarge	1 x 800 GB SSD
i2.2xlarge	2 x 800 GB SSD (1600 GB)
i2.4xlarge	4 x 800 GB SSD (3200 GB)
i2.8xlarge	8 x 800 GB SSD (6400 GB)
m1.small	1 x 160 GB
m1.medium	1 x 410 GB
m1.large	2 x 420 GB (840 GB)
m1.xlarge	4 x 420 GB (1680 GB)
m2.xlarge	1 x 420 GB
m2.2xlarge	1 x 850 GB
m2.4xlarge	2 x 840 GB (1680 GB)
m3.medium	1 x 4 GB SSD
m3.large	1 x 32 GB SSD
m3.xlarge	2 x 40 GB SSD (80 GB)
m3.2xlarge	2 x 80 GB SSD (160 GB)
r3.large	1 x 32 GB SSD
r3.xlarge	1 x 80 GB SSD
r3.2xlarge	1 x 160 GB SSD
r3.4xlarge	1 x 320 GB SSD
r3.8xlarge	2 X 320 GB SSD (640 GB)

Add Instance Store Volumes to Your EC2 Instance

You specify the EBS volumes and instance store volumes for your instance using a block device mapping. Each entry in a block device mapping includes a device name and the volume that it maps to. The default block device mapping is specified by the AMI you use. Alternatively, you can specify a block device mapping for the instance when you launch it. For more information, see Block Device Mapping (p. 587).

A block device mapping always specifies the root volume for the instance. The root volume is either an Amazon EBS volume or an instance store volume. For more information, see Storage for the Root Device (p. 53). The root volume is mounted automatically. For instances with an instance store volume for the root volume, the size of this volume varies by AMI, but the maximum size is 10 GiB.

You can use a block device mapping to specify additional EBS volumes when you launch your instance, or you can attach additional EBS volumes after your instance is running. For more information, see Amazon EBS Volumes (p. 518).

You can specify the instance store volumes for your instance only when you launch an instance. You can't attach instance store volumes to an instance after you've launched it.

The number and size of available instance store volumes for your instance varies by instance type. Some instance types do not support instance store volumes. For more information about the instance store volumes support by each instance type, see Instance Store Volumes (p. 578). If the instance type you choose for your instance supports instance store volumes, you must add them to the block device mapping for the instance when you launch it. After you launch the instance, you must ensure that the instance store volumes for your instance are formatted and mounted before you can use them. Note that the root volume of an instance store-backed instance is mounted automatically.

Contents

- Adding Instance Store Volumes to an AMI (p. 580)
- Adding Instance Store Volumes to an Instance (p. 581)
- Making Instance Store Volumes Available on Your Instance (p. 582)

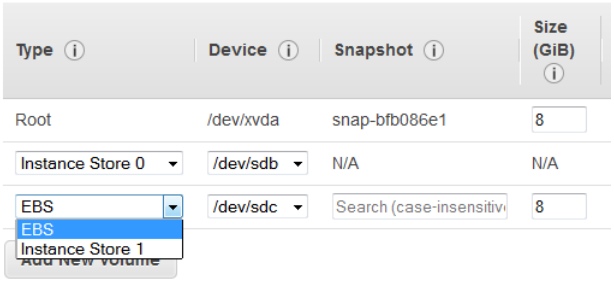
Adding Instance Store Volumes to an AMI

You can create an AMI with a block device mapping that includes instance store volumes. After you add instance store volumes to an AMI, any instance that you launch from the AMI includes these instance store volumes. Note that when you launch an instance, you can omit volumes specified in the AMI block device mapping and add new volumes.

> **Important**
> For M3 instances, you must specify instance store volumes in the block device mapping for the instance when you launch it. When you launch an M3 instance, instance store volumes specified in the block device mapping for the AMI may be ignored if they are not specified as part of the instance block device mapping.

To add instance store volumes to an Amazon EBS-backed AMI using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Instances**.
3. Select an instance, click **Actions**, select **Image**, and then select **Create Image**.
4. In the **Create Image** dialog, add a meaningful name and description for your image.
5. For each instance store volume to add, click **Add New Volume**, select an instance store volume from **Type**, and select a device name from **Device**. (For more information, see Device Naming on Windows Instances (p. 586).) The number of available instance store volumes depends on the instance type.



6. Click **Create Image**.

To add instance store volumes to an AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

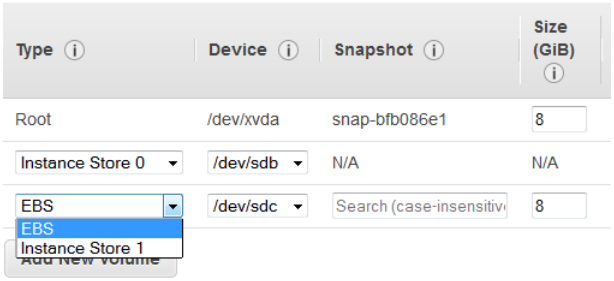
- create-image or register-image (AWS CLI)
- ec2-create-image ec2-register (Amazon EC2 CLI)

Adding Instance Store Volumes to an Instance

When you launch an instance, the default block device mapping is provided by the specified AMI. If you need additional instance store volumes, you must add them to the instance as you launch it. Note that you can also omit devices specified in the AMI block device mapping.

To update the block device mapping for an instance using the console

1. Open the Amazon EC2 console.
2. From the dashboard, click **Launch Instance**.
3. In **Step 1: Choose an Amazon Machine Image (AMI)**, choose the AMI to use and click **Select**.
4. Follow the wizard to complete **Step 1: Choose an Amazon Machine Image (AMI)**, **Step 2: Choose an Instance Type**, and **Step 3: Configure Instance Details**.
5. In **Step 4: Add Storage**, modify the existing entries as needed. For each instance store volume to add, click **Add New Volume**, select an instance store volume from **Type**, and select a device name from **Device**. The number of available instance store volumes depends on the instance type.



6. Complete the wizard to launch the instance.

To update the block device mapping for an instance using the command line

You can use one of the following options commands with the corresponding command. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- `--block-device-mappings` with run-instances (AWS CLI)
- `--block-device-mapping` with ec2-run-instances (Amazon EC2 CLI)
- `-BlockDeviceMapping` with New-EC2Instance (AWS Tools for Windows PowerShell)

Making Instance Store Volumes Available on Your Instance

After you launch an instance, the instance store volumes are available to the instance, but you can't access them until they are mounted. For Linux instances, the instance type determines which instance store volumes are mounted for you and which are available for you to mount yourself. For Windows instances, the EC2Config service mounts the instance store volumes for an instance. The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different than the name that Amazon EC2 recommends.

Many instance store volumes are pre-formatted with the ext3 file system. SSD-based instance store volumes that support TRIM instruction are not pre-formatted with any file system. However, you can format volumes with the file system of your choice after you launch your instance. For more information, see Instance Store Volume TRIM Support (p. 582). For Windows instances, the EC2Config service reformats the instance store volumes with the NTFS file system.

You can confirm that the instance store devices are available from within the instance itself using instance metadata. For more information, see Viewing the Instance Block Device Mapping for Instance Store Volumes (p. 596).

For Windows instances, you can also view the instance store volumes using Windows Disk Management. For more information, see Listing the Disks Using Windows Disk Management (p. 597).

SSD Instance Store Volumes

The following instances support instance store volumes that use solid state drives (SSD) to deliver very high random I/O performance: C3, G2, HI1, I2, M3, and R3. For more information about the instance store volumes support by each instance type, see Instance Store Volumes (p. 578).

Like other instance store volumes, you must map the SSD instance store volumes for your instance when you launch it, and the data on an SSD instance volume persists only for the life of its associated instance. For more information, see Add Instance Store Volumes to Your EC2 Instance (p. 579).

Instance Store Volume TRIM Support

The following instances support SSD volumes with TRIM: I2, R3.

> **Important**
> Instances running Windows Server 2012 R2 support TRIM as of AWS PV Driver version 7.3.0. Instances running earlier versions of Windows Server do not support TRIM.

With instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller when you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information about using TRIM commands, see the documentation for the operating system for your instance.

Instance store volumes that support TRIM are fully trimmed before they are allocated to your instance. These volumes are not formatted with a file system when an instance launches, so you must format them before they can be mounted and used. For faster access to these volumes, you should specify the file system-specific option that skips the TRIM operation when you format them. On Linux, you should also add the `discard` option to your mount command or `/etc/fstab` file entries for the devices that support TRIM so that they use this feature effectively. On Windows, use the following command: `fsutil behavior set DisableDeleteNotify 1`.

Amazon Simple Storage Service (Amazon S3)

Amazon S3 is a repository for Internet data. Amazon S3 provides access to reliable, fast, and inexpensive data storage infrastructure. It is designed to make web-scale computing easy by enabling you to store and retrieve any amount of data, at any time, from within Amazon EC2 or anywhere on the web. Amazon S3 stores data objects redundantly on multiple devices across multiple facilities and allows concurrent read or write access to these data objects by many separate clients or application threads. You can use the redundant data stored in Amazon S3 to recover quickly and reliably from instance or application failures.

Amazon EC2 uses Amazon S3 for storing Amazon Machine Images (AMIs). You use AMIs for launching EC2 instances. In case of instance failure, you can use the stored AMI to immediately launch another instance, thereby allowing for fast recovery and business continuity.

Amazon EC2 also uses Amazon S3 to store snapshots (backup copies) of the data volumes. You can use snapshots for recovering data quickly and reliably in case of application or system failures. You can also use snapshots as a baseline to create multiple new data volumes, expand the size of an existing data volume, or move data volumes across multiple Availability Zones, thereby making your data usage highly scalable. For more information about using data volumes and snapshots, see Amazon Elastic Block Store (p. 516).

Objects are the fundamental entities stored in Amazon S3. Every object stored in Amazon S3 is contained in a bucket. Buckets organize the Amazon S3 namespace at the highest level and identify the account responsible for that storage. Amazon S3 buckets are similar to Internet domain names. Objects stored in the buckets have a unique key value and are retrieved using a HTTP URL address. For example, if an object with a key value `/photos/mygarden.jpg` is stored in the `myawsbucket` bucket, then it is addressable using the URL `http://myawsbucket.s3.amazonaws.com/photos/mygarden.jpg`.

For more information about the features of Amazon S3, see the Amazon S3 product page.

Amazon S3 and Amazon EC2

Given the benefits of Amazon S3 for storage, you may decide to use this service to store files and data sets for use with EC2 instances. There are several ways to move data to and from Amazon S3 to your instances. In addition to the examples discussed below, there are a variety of tools that people have written that you can use to access your data in Amazon S3 from your computer or your instance. Some of the common ones are discussed in the AWS forums.

If you have permission, you can copy a file to or from Amazon S3 and your instance using one of the following methods.

GET or wget

The **wget** utility is an HTTP and FTP client that allows you to download public objects from Amazon S3. It is installed by default in Amazon Linux and most other distributions, and available for download on Windows. To download an Amazon S3 object, use the following command, substituting the URL of the object to download.

```
wget http://s3.amazonaws.com/my_bucket/my_folder/my_file.ext
```

This method requires that the object you request is public; if the object is not public, you receive an `ERROR 403: Forbidden` message. If you receive this error, open the Amazon S3 console and change the permissions of the object to public. For more information, see the Amazon Simple Storage Service Developer Guide.

AWS Command Line Interface

The AWS Command Line Interface (AWS CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts. The AWS CLI allows users to authenticate themselves and download restricted items from Amazon S3 and also to upload items. For more information, such as how to install and configure the tools, see the AWS Command Line Interface detail page.

The **aws s3 cp** command is similar to the Unix **cp** command (the syntax is: **aws s3 cp *source destination***). You can copy files from Amazon S3 to your instance, you can copy files from your instance to Amazon S3, and you can even copy files from one Amazon S3 location to another.

Use the following command to copy an object from Amazon S3 to your instance.

```
C:\> aws s3 cp s3://my_bucket/my_folder/my_file.ext my_copied_file.ext
```

Use the following command to copy an object from your instance back into Amazon S3.

```
C:\> aws s3 cp my_copied_file.ext s3://my_bucket/my_folder/my_file.ext
```

Use the following command to copy an object from one Amazon S3 location to another.

```
C:\> aws s3 cp s3://my_bucket/my_folder/my_file.ext s3://my_buck
et/my_folder/my_file2.ext
```

The **aws s3 sync** command can synchronize an entire Amazon S3 bucket to a local directory location. This can be helpful for downloading a data set and keeping the local copy up-to-date with the remote set. The command syntax is: **aws s3 sync *source destination***. If you have the proper permissions on the Amazon S3 bucket, you can push your local directory back up to the cloud when you are finished by reversing the source and destination locations in the command.

Use the following command to download an entire Amazon S3 bucket to a local directory on your instance.

```
C:\> aws s3 sync s3://remote_S3_bucket local_directory
```

AWS Tools for Windows PowerShell

Windows instances have the benefit of a graphical browser that you can use to access the Amazon S3 console directly; however, for scripting purposes, Windows users can also use the AWS Tools for Windows PowerShell to move objects to and from Amazon S3.

Use the following command to copy an Amazon S3 object to your Windows instance.

```
PS C:\> Copy-S3Object -BucketName my_bucket -Key my_folder/my_file.ext -LocalFile
 my_copied_file.ext
```

Amazon S3 API

If you are a developer, you can use an API to access data in Amazon S3. For more information, see the Amazon Simple Storage Service Developer Guide. You can use this API and its examples to help develop your application and integrate it with other APIs and SDKs, such as the `boto` Python interface.

Instance Volume Limits

The maximum number of volumes that your instance can have depends on the operating system. When considering how many volumes to add to your instance, you should consider whether you need increased I/O bandwidth or increased storage capacity.

Contents

- Linux-Specific Volume Limits (p. 585)
- Windows-Specific Volume Limits (p. 585)
- Bandwidth vs Capacity (p. 585)

Linux-Specific Volume Limits

Attaching more than 40 volumes can cause boot failures. Note that this number includes the root volume, plus any attached instance store volumes and EBS volumes. If you experience boot problems on an instance with a large number of volumes, stop the instance, detach any volumes that are not essential to the boot process, and then reattach the volumes after the instance is running.

> **Important**
> Attaching more than 40 volumes to a Linux instance is supported on a best effort basis only and is not guaranteed.

Windows-Specific Volume Limits

The following table shows the volume limits for Windows instances based on the driver used. Note that these numbers include the root volume, plus any attached instance store volumes and EBS volumes.

> **Important**
> Attaching more than the following volumes to a Windows instance is supported on a best effort basis only and is not guaranteed.

Driver	Volume Limit
AWS PV	26
Citrix PV	26
Red Hat PV	17

We do not recommend that you give a Windows instance more than 26 volumes with AWS PV or Citrix PV drivers, as it is likely to cause performance issues.

To determine which PV drivers your instance is using, or to upgrade your Windows instance from Red Hat to Citrix PV drivers, see Upgrading PV Drivers on Your Windows AMI (p. 265).

For more information about how device names related to volumes, see Mapping Disks to Volumes on Your Windows EC2 Instance (p. 596).

Bandwidth vs Capacity

For consistent and predictable bandwidth use cases, use EBS-optimized or 10 Gigabit network connectivity instances and General Purpose (SSD) or Provisioned IOPS (SSD) volumes. Follow the guidance in Amazon EC2 Instance Configuration (p. 562) to match the IOPS you have provisioned for your volumes to the bandwidth available from your instances for maximum performance. For RAID configurations, many

administrators find that arrays larger than 8 volumes have diminished performance returns due to increased I/O overhead. Test your individual application performance and tune it as required.

Device Naming on Windows Instances

When you attach a volume to your instance, you include a device name for the volume. This device name is used by Amazon EC2. The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different from the name that Amazon EC2 uses.

Contents

- Available Device Names (p. 586)
- Device Name Considerations (p. 586)

For information about device names on Linux instances, see Device Naming on Linux Instances in the *Amazon EC2 User Guide for Linux Instances*.

Available Device Names

The following table lists the available device names for Windows instances. The number of volumes that you can attach to your instance is determined by the operating system. For more information, see Instance Volume Limits (p. 585).

Xen Driver Type	Available	Reserved for Root	Used for Instance Store Volumes	Recommended for EBS Volumes
AWS PV, Citrix PV	xvd[a-z] xvd[b-c][a-z] /dev/sda1 /dev/sd[b-e]	/dev/sda1	xvd[a-e] xvdc[a-x] (hs1.8xlarge)	xvd[f-z]
Red Hat PV	xvd[a-z] xvd[b-c][a-z] /dev/sda1 /dev/sd[b-e]	/dev/sda1	xvd[a-e] xvdc[a-x] (hs1.8xlarge)	xvd[f-p]

Note that you can determine the root device name for your particular AMI with the following AWS CLI command:

```
aws ec2 describe-images --image-ids image_id --query Images[].RootDeviceName
```

For more information about instance store volumes, see Amazon EC2 Instance Store (p. 577). For information about the root device storage, see Root Device Volume (p. 8).

Device Name Considerations

Keep the following in mind when selecting a device name:

- Although you can attach your EBS volumes using the device names used to attach instance store volumes, we strongly recommend that you don't because the behavior can be unpredictable.
- Amazon EC2 Windows AMIs come with an additional service installed, the **Ec2Config Service**. The Ec2Config service runs as a local system and performs various functions to prepare an instance when it first boots up. After the devices have been mapped with the drives, the Ec2Config service then initializes and mounts the drives. The root drive is initialized and mounted as `c:\`. The instance store volumes that come attached to the instance are initialized and mounted as `d:\`, `e:\`, and so on. By default, when an EBS volume is attached to a Windows instance, it can show up as any drive letter on the instance. You can change the settings of the Ec2Config service to set the drive letters of the EBS volumes per your specifications. For more information, see Configuring a Windows Instance Using the EC2Config Service (p. 235) and Mapping Disks to Volumes on Your Windows EC2 Instance (p. 596).

Block Device Mapping

Each instance that you launch has an associated root device volume, either an Amazon EBS volume or an instance store volume. You can use block device mapping to specify additional EBS volumes or instance store volumes to attach to an instance when it's launched. You can also attach additional EBS volumes to a running instance; see Attaching an Amazon EBS Volume to an Instance (p. 529). However, the only way to attach instance store volumes to an instance is to use block device mapping to attach them as the instance is launched.

For more information about root device volumes, see Root Device Volume (p. 8).

Contents

- Block Device Mapping Concepts (p. 587)
- AMI Block Device Mapping (p. 590)
- Instance Block Device Mapping (p. 592)

Block Device Mapping Concepts

A *block device* is a storage device that moves data in sequences of bytes or bits (blocks). These devices support random access and generally use buffered I/O. Examples include hard disks, CD-ROM drives, and flash drives. A block device can be physically attached to a computer or accessed remotely as if it were physically attached to the computer. Amazon EC2 supports two types of block devices:

- Instance store volumes (virtual devices whose underlying hardware is physically attached to the host computer for the instance)
- EBS volumes (remote storage devices)

A *block device mapping* defines the block devices (instance store volumes and EBS volumes) to attach to an instance. You can specify a block device mapping as part of creating an AMI so that the mapping is used by all instances launched from the AMI. Alternatively, you can specify a block device mapping when you launch an instance, so this mapping overrides the one specified in the AMI from which you launched the instance.

Block Device Mapping Entries

When you create a block device mapping, you specify the following information for each block device that you need to attach to the instance:

- The device name used within Amazon EC2. For more information, see Device Naming on Windows Instances (p. 586).

Important
The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different from the name that Amazon EC2 recommends.

- [Instance store volumes] The virtual device: `ephemeral[0-23]`. Note that the number and size of available instance store volumes for your instance varies by instance type.
- [EBS volumes] The ID of the snapshot to use to create the block device (snap-*xxxxxxxx*). This value is optional as long as you specify a volume size.
- [EBS volumes] The size of the volume, in GiB. The specified size must be greater than or equal to the size of the specified snapshot.
- [EBS volumes] Whether to delete the volume on instance termination (`true` or `false`). The default value is `true`.
- [EBS volumes] The volume type, which can be `gp2` for General Purpose (SSD) volumes, `standard` for Magnetic volumes or `io1` for Provisioned IOPS (SSD) volumes. The default value is `gp2` for General Purpose (SSD) volumes in the Amazon EC2 console, and `standard` for Magnetic volumes in the AWS SDKs, the AWS CLI, or the Amazon EC2 CLI.
- [EBS volumes] The number of input/output operations per second (IOPS) that the volume supports. (Not used with General Purpose (SSD) or Magnetic volumes.)

Block Device Mapping Instance Store Caveats

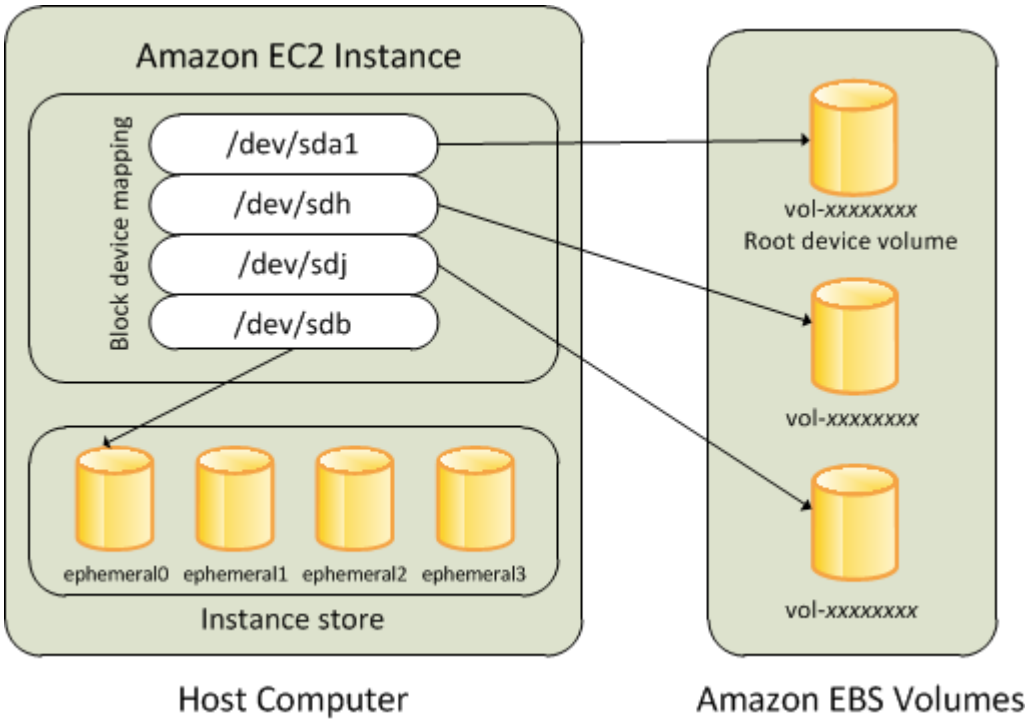
There are several caveats to consider when launching instances with AMIs that have instance store volumes in their block device mappings.

- Some instance types include more instance store volumes than others, and some instance types contain no instance store volumes at all. If your instance type supports one instance store volume, and your AMI has mappings for two instance store volumes, then the instance launches with one instance store volume.
- Instance store volumes can only be mapped at launch time. You cannot stop an instance without instance store volumes (such as the `t2.micro`), change the instance to a type that supports instance store volumes, and then restart the instance with instance store volumes. However, you can create an AMI from the instance and launch it on an instance type that supports instance store volumes, and map those instance store volumes to the instance.
- If you launch an instance with instance store volumes mapped, and then stop the instance and change it to an instance type with fewer instance store volumes and restart it, the instance store volume mappings from the initial launch still show up in the instance metadata. However, only the maximum number of supported instance store volumes for that instance type are available to the instance.

  Note
  When an instance is stopped, all data on the instance store volumes is lost.
- Depending on instance store capacity at launch time, M3 instances may ignore AMI instance store block device mappings at launch unless they are specified at launch. You should specify instance store block device mappings at launch time, even if the AMI you are launching has the instance store volumes mapped in the AMI, to ensure that the instance store volumes are available when the instance launches.

Example Block Device Mapping

This figure shows an example block device mapping for an EBS-backed instance. It maps `/dev/sdb` to `ephemeral0` and maps two EBS volumes, one to `/dev/sdh` and the other to `/dev/sdj`. It also shows the EBS volume that is the root device volume, `/dev/sda1`.



Note that this example block device mapping is used in the example commands and APIs in this topic. You can find example commands and APIs that create block device mappings here:

- Specifying a Block Device Mapping for an AMI (p. 590)
- Updating the Block Device Mapping when Launching an Instance (p. 592)

How Devices Are Made Available in the Operating System

Device names like `/dev/sdh` and `xvdh` are used by Amazon EC2 to describe block devices. The block device mapping is used by Amazon EC2 to specify the block devices to attach to an EC2 instance. After a block device is attached to an instance, it must be mounted by the operating system before you can access the storage device. When a block device is detached from an instance, it is unmounted by the operating system and you can no longer access the storage device.

With a Windows instance, the device names specified in the block device mapping are mapped to their corresponding block devices when the instance first boots, and then the Ec2Config service initializes and mounts the drives. The root device volume is mounted as `C:\`. The instance store volumes are mounted as `D:\`, `E:\`, and so on. When an EBS volume is mounted, it can be mounted using any available drive letter. However, you can configure how the Ec2Config Service assigns drive letters to EBS volumes; for more information, see Configuring a Windows Instance Using the EC2Config Service (p. 235).

Viewing Block Device Mappings

You can view information about each block device in a block device mapping. For details, see:

- Viewing the EBS Volumes in an AMI Block Device Mapping (p. 592)
- Viewing the EBS Volumes in an Instance Block Device Mapping (p. 595)
- Viewing the Instance Block Device Mapping for Instance Store Volumes (p. 596)

AMI Block Device Mapping

Each AMI has a block device mapping that specifies the block devices to attach to an instance when it is launched from the AMI. An AMI that Amazon provides includes a root device only. To add more block devices to an AMI, you must create your own AMI.

Contents

- Specifying a Block Device Mapping for an AMI (p. 590)
- Viewing the EBS Volumes in an AMI Block Device Mapping (p. 592)

Specifying a Block Device Mapping for an AMI

There are two ways to specify volumes in addition to the root volume when you create an AMI. If you've already attached volumes to a running instance before you create an AMI from the instance, the block device mapping for the AMI includes those same volumes. For EBS volumes, the existing data is saved to a new snapshot, and it's this new snapshot that's specified in the block device mapping. For instance store volumes, the data is not preserved.

For an EBS-backed AMI, you can add EBS volumes and instance store volumes using a block device mapping. For an instance store-backed AMI, you can add only instance store volumes using a block device mapping.

> **Note**
> For M3 instances, you must specify instance store volumes in the block device mapping for the instance when you launch it. When you launch an M3 instance, instance store volumes specified in the block device mapping for the AMI may be ignored if they are not specified as part of the instance block device mapping.

To add volumes to an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**.
3. Select an instance, click **Actions**, select **Image**, and then select **Create Image**.
4. In the **Create Image** dialog box, click **Add New Volume**.
5. Select a volume type from the **Type** list and a device name from the **Device** list. For an EBS volume, you can optionally specify a snapshot, volume size, and volume type.
6. Click **Create Image**.

To add volumes to an AMI using the AWS CLI

Use the create-image command to specify a block device mapping for an EBS-backed AMI. Use the register-image command to specify a block device mapping for an instance store-backed AMI.

Specify the block device mapping using the following parameter:

```
--block-device-mappings [mapping, ...]
```

To add an instance store volume, use the following mapping:

```
{
    "DeviceName": "xvdb",
    "VirtualName": "ephemeral0"
}
```

To add an empty 100 GiB Magnetic volume, use the following mapping:

```
{
    "DeviceName": "xvdg",
    "Ebs": {
      "VolumeSize": 100
    }
}
```

To add an EBS volume based on a snapshot, use the following mapping:

```
{
    "DeviceName": "xvdh",
    "Ebs": {
      "SnapshotId": "snap-xxxxxxxx"
    }
}
```

To omit a mapping for a device, use the following mapping:

```
{
    "DeviceName": "xvdj",
    "NoDevice": ""
}
```

To add volumes to an AMI using the Amazon EC2 CLI

Use the ec2-create-image command to specify a block device mapping for an EBS-backed AMI. Use the ec2-register command to specify a block device mapping for an instance store-backed AMI.

Specify the block device mapping using the following parameter:

```
-b "devicename=blockdevice"
```

devicename
The device name within Amazon EC2.

blockdevice
To omit a mapping for the device from the AMI, specify `none`.

To add an instance store volume, specify `ephemeral[0..23]`.

To add an EBS volume to an EBS-backed instance, specify [*snapshot-id*]:[*size*]:[*delete-on-termination*]:[*type*[:*iops*]]

- To add an empty volume, omit the snapshot ID and specify a volume size instead.
- To indicate whether the volume should be deleted on termination, specify `true` or `false`; the default value is `true`.
- To create a Provisioned IOPS (SSD) volume, specify `io1` and to create a General Purpose (SSD) volume, specify `gp2`; the default type is `standard` for Magnetic volumes. If the type is `io1`, you can also provision the number of IOPS the volume supports.

You can specify multiple block devices in a single command using multiple `-b` parameters. For example, the following parameters add an instance store volume as `xvdb`, an EBS volume based on a snapshot as `xvdh`, and an empty 100 GiB EBS volume as `xvdj`.

```
-b "xvdb=ephemeral0" -b "xvdh=snap-d5eb27ab" -b "xvdj=:100"
```

Viewing the EBS Volumes in an AMI Block Device Mapping

You can easily enumerate the EBS volumes in the block device mapping for an AMI.

To view the EBS volumes for an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. Select **EBS images** from the **Filter** drop-down list to get a list of EBS-backed AMIs.
4. Select the desired AMI, and look at the **Details** tab. At a minimum, the following information is available for the root device:

   - **Root Device Type** (`ebs`)
   - **Root Device Name** (for example, `/dev/sda1`)
   - **Block Devices** (for example, `/dev/sda1=snap-e1eb279f:8:true`)

   If the AMI was created with additional EBS volumes using a block device mapping, the **Block Devices** field displays the mapping for those additional volumes as well. (Recall that this screen doesn't display instance store volumes.)

To view the EBS volumes for an AMI using the AWS CLI

Use the describe-images command to enumerate the EBS volumes in the block device mapping for an AMI.

To view the EBS volumes for an AMI using the Amazon EC2 CLI

Use the ec2-describe-images command to enumerate the EBS volumes in the block device mapping for an AMI.

Instance Block Device Mapping

By default, an instance that you launch includes any storage devices specified in the block device mapping of the AMI from which you launched the instance. You can specify changes to the block device mapping for an instance when you launch it, and these updates overwrite or merge with the block device mapping of the AMI. However, you can only modify the volume size, volume type, and **Delete on Termination** flag on the block device mapping entry for the root device volume.

Contents

- Updating the Block Device Mapping when Launching an Instance (p. 592)
- Viewing the EBS Volumes in an Instance Block Device Mapping (p. 595)
- Viewing the Instance Block Device Mapping for Instance Store Volumes (p. 596)

Updating the Block Device Mapping when Launching an Instance

You can add EBS volumes and instance store volumes to an instance when you launch it. Note that updating the block device mapping for an instance doesn't make a permanent change to the block device mapping of the AMI from which it was launched.

To add volumes to an instance using the console

1. Open the Amazon EC2 console.
2. From the dashboard, click **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, choose the AMI to use and click **Select**.
4. Follow the wizard to complete the **Choose an Instance Type** and **Configure Instance Details** pages.
5. On the **Add Storage** page, you can modify the root volume, EBS volumes, and instance store volumes as follows:
   - To change the size of the root volume, locate the **Root** volume under the **Type** column, and change its **Size** field.
   - To suppress an EBS volume specified by the block device mapping of the AMI used to launch the instance, locate the volume and click its **Delete** icon.
   - To add an EBS volume, click **Add New Volume**, select **EBS** from the **Type** list, and fill in the fields (**Device**, **Snapshot**, and so on).
   - To suppress an instance store volume specified by the block device mapping of the AMI used to launch the instance, locate the volume, and click its **Delete** icon.
   - To add an instance store volume, click **Add New Volume**, select **Instance Store** from the **Type** list, and select a device name from **Device**.
6. Complete the remaining wizard pages, and then click **Launch**.

To add volumes to an instance using the AWS CLI

Use the run-instances command to specify a block device mapping for an instance.

Specify the block device mapping using the following parameter:

```
--block-device-mappings [mapping, ...]
```

For example, suppose that an EBS-backed AMI specifies the following block device mapping:

- xvdb=ephemeral0
- xvdh=snap-92d333fb
- xvdj=:100

To prevent `xvdj` from attaching to an instance launched from this AMI, use the following mapping:

```
{
    "DeviceName": "xvdj",
    "NoDevice": ""
}
```

To increase the size of `xvdh` to 300 GiB, specify the following mapping. Notice that you don't need to specify the snapshot ID for `xvdh`, because specifying the device name is enough to identify the volume.

```
{
    "DeviceName": "xvdh",
    "Ebs": {
      "VolumeSize": 300
```

```
    }
}
```

To attach an additional instance store volume, `xvdc`, specify the following mapping. If the instance type doesn't support multiple instance store volumes, this mapping has no effect.

```
{
    "DeviceName": "xvdc",
    "VirtualName": "ephemeral1"
}
```

To add volumes to an instance using the Amazon EC2 CLI

Use the ec2-run-instances command to specify a block device mapping for an instance.

Specify the block device mapping using the following parameter:

```
-b "devicename=blockdevice"
```

devicename
The device name within Amazon EC2.

blockdevice
To omit a mapping for the device from the AMI, specify `none`.

To add an instance store volume, specify `ephemeral[0..23]`.

To add an EBS volume to an EBS-backed instance, specify [*snapshot-id*]:[*size*]:[*delete-on-termination*]:[*type*[:*iops*]].

- To add an empty EBS volume, omit the snapshot ID and specify a volume size instead.
- To indicate whether the EBS volume is deleted on termination, specify `true` or `false`; the default value is `true`.
- To create a Provisioned IOPS (SSD) volume, specify `io1` and to create a General Purpose (SSD) volume, specify `gp2`; the default type is `standard` for Magnetic volumes. If the type is `io1`, you can also provision the number of IOPS the volume supports.

For example, suppose that an EBS-backed AMI specifies the following block device mapping:

- xvdb=ephemeral0
- xvdh=snap-92d333fb
- xvdj=:100

To prevent `xvdj` from attaching to an instance launched from this AMI, use the following option:

```
-b "xvdj=none"
```

To increase the size of `xvdh` to 300 GiB, use the following option:

```
-b "xvdh=:300"
```

Notice that you didn't need to specify the snapshot ID for `xvdh`, because specifying the device name is enough to identify the volume.

To attach an additional instance store volume, `xvdc`, use the following option. If the instance type doesn't support multiple instance store volumes, this option has no effect.

```
-b "xvdc=ephemeral1"
```

Viewing the EBS Volumes in an Instance Block Device Mapping

You can easily enumerate the EBS volumes mapped to an instance.

Note
For instances launched before the release of the 2009-10-31 API, AWS can't display the block device mapping. You must detach and reattach the volumes so that AWS can display the block device mapping.

To view the EBS volumes for an instance using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**.
3. In the search bar, type **Root Device Type**, and then select **EBS**. This displays a list of EBS-backed instances.
4. Locate and click the desired instance and look at the details displayed in the **Description** tab. At a minimum, the following information is available for the root device:

   - **Root device type** (`ebs`)
   - **Root device** (for example, `/dev/sda1`)
   - **Block devices** (for example, `/dev/sda1`, `xvdh`, and `xvdj`)

   If the instance was launched with additional EBS volumes using a block device mapping, the **Block devices** box displays those additional volumes as well as the root device. (Recall that this dialog box doesn't display instance store volumes.)

5. To display additional information about a block device, click its entry next to **Block devices**. This displays the following information for the block device:

   - **EBS ID** (vol-*xxxxxxxx*)
   - **Root device type** (`ebs`)
   - **Attachment time** (*yyyy-mm*T*hh*:*mm*:*ss*.*ss*TZD)
   - **Block device status** (`attaching`, `attached`, `detaching`, `detached`)
   - **Delete on termination** (`Yes`, `No`)

To view the EBS volumes for an instance using the AWS CLI

Use the describe-instances command to enumerate the EBS volumes in the block device mapping for an instance.

To view the EBS volumes for an instance using the Amazon EC2 CLI

Use the ec2-describe-instances command to enumerate the EBS volumes in the block device mapping for an instance.

Viewing the Instance Block Device Mapping for Instance Store Volumes

When you view the block device mapping for your instance, you can see only the EBS volumes, not the instance store volumes. You can use instance metadata to query the complete block device mapping. The base URI for all requests for instance metadata is `http://169.254.169.254/latest/`.

First, connect to your running instance. For Windows instances, install wget on the instance if it is not installed already.

Use this query on a running instance to get its block device mapping.

```
C:\> wget http://169.254.169.254/latest/meta-data/block-device-mapping/
```

The response includes the names of the block devices for the instance. For example, the output for an instance store–backed `m1.small` instance looks like this.

```
ami
ephemeral0
root
swap
```

The ami device is the root device as seen by the instance. The instance store volumes are named `ephemeral[0-23]`. The swap device is for the page file. If you've also mapped EBS volumes, they appear as ebs1, ebs2, and so on.

To get details about an individual block device in the block device mapping, append its name to the previous query, as shown here.

```
C:\> wget http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

For more information, see Instance Metadata and User Data (p. 160).

Mapping Disks to Volumes on Your Windows EC2 Instance

Your Windows EC2 instance comes with an EBS volume that serves as the root volume. If your Windows instance uses AWS PV or Citrix PV drivers, you can optionally add up to 25 volumes, making a total of 26 volumes. For more information, see Instance Volume Limits (p. 585)

Depending on the instance type of your instance, you'll have from 0 to 24 possible instance store volumes available to the instance. To use any of the instance store volumes that are available to your instance, you must specify them when you create your AMI or launch your instance. You can also add EBS volumes when you create your AMI or launch your instance, or attach them while your instance is running.

When you add a volume to your instance, you specify the device name that Amazon EC2 uses. For more information, see Device Naming on Windows Instances (p. 586). AWS Windows Amazon Machine Images

(AMIs) contain a set of drivers that are used by Amazon EC2 to map instance store and EBS volumes to Windows disks and drive letters. If you launch an instance from a Windows AMI that uses Citrix paravirtualized (PV) or AWS PV drivers, you can use the relationships described on this page to map your Windows disks to your instance store and EBS volumes. If your Windows AMI uses Red Hat PV drivers, you can update your instance to use the Citrix drivers. For more information, see Upgrading PV Drivers on Your Windows AMI (p. 265).

Contents

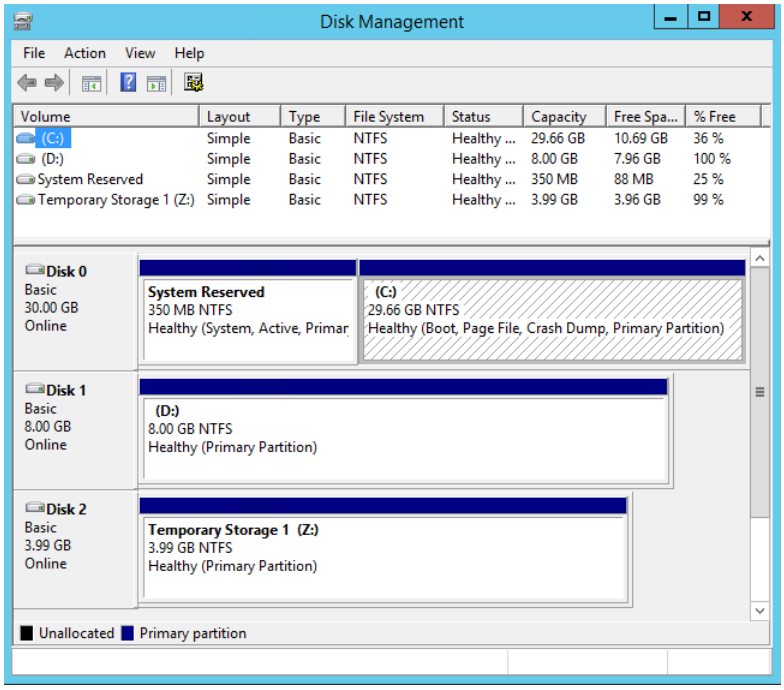
- Listing the Disks Using Windows Disk Management (p. 597)
- Listing the Disks Using Windows PowerShell (p. 598)
- Disk Device to Device Name Mapping (p. 600)

Listing the Disks Using Windows Disk Management

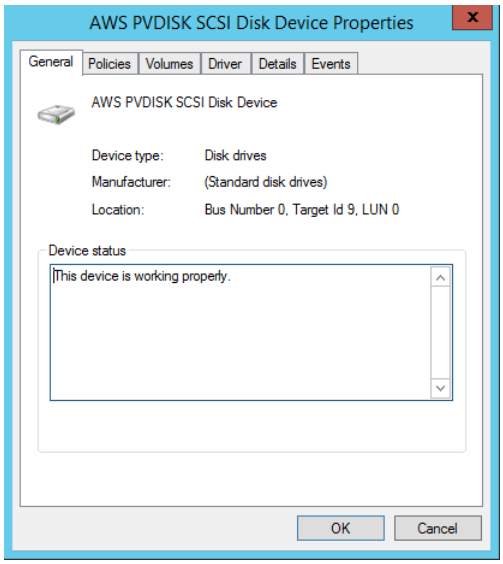
You can find the disks on your Windows instance using Windows Disk Management.

To find the disks on your Windows instance

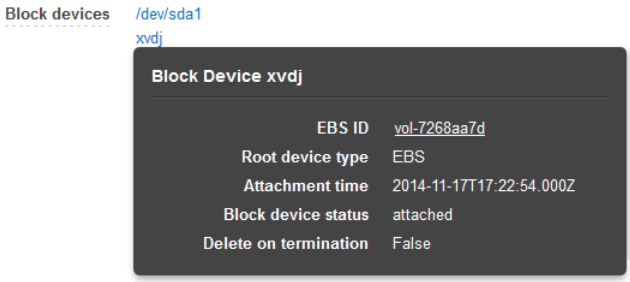
1. Log in to your Windows instance using Remote Desktop. For more information, see, Connecting to Your Windows Instance Using RDP (p. 216).
2. Start the Disk Management utility. On Windows Server 2012, on the taskbar, right-click the Windows logo, and then select **Disk Management**. On Windows Server 2008, click **Start**, point to **Administrative Tools**, select **Computer Management**, and then select **Disk Management**.
3. Review the disks. Disk 0 is the root volume, which is an EBS volume mounted as `C:\`. If there are no other disks shown, then your instance does not come with instance store volumes, and you didn't specify any EBS volumes when you created the AMI or launched the instance. Otherwise, you'll see additional disks. For example, the following disks are available if you launch an `m3.medium` instance with an additional empty EBS volume. Disk 1 is the EBS volume, and Disk 2 is the instance store volume.



4. Right-click the gray pane labeled Disk 1, and then select **Properties**. Note the value of **Location** and look it up in the tables in Disk Device to Device Name Mapping (p. 600). For example, the following disk has the location `Bus Number 0, Target Id 9, LUN 0`. According to the table for EBS volumes, the device name for this location is `xvdj`.



5. To map the device name of an EBS volume to its volume ID, open the Amazon EC2 console on your computer. In the navigation pane, select **Instances**, and then select your instance. Under **Block devices**, click the device name, and locate **EBS ID**. For this example, the volume ID is `vol-7268aa7d`.



Note that the Amazon EC2 console shows only the EBS volumes.

Listing the Disks Using Windows PowerShell

The following PowerShell script lists each disk and its corresponding device name and volume.

```
# List the Windows disks

# Create a hash table that maps each device to a SCSI target
$Map = @{"0" = '/dev/sda1'}
for($x = 1; $x -le 26; $x++) {$Map.add($x.ToString(),
[String]::Format("xvd{0}",[char](97 + $x)))}
for($x = 78; $x -le 102; $x++) {$Map.add($x.ToString(),
[String]::Format("xvdc{0}",[char](19 + $x)))}

Try {
    # Use the metadata service to discover which instance the script is running
```

```
 on
   $InstanceId = (Invoke-WebRequest '169.254.169.254/latest/meta-data/instance-
id').Content
   $AZ = (Invoke-WebRequest '169.254.169.254/latest/meta-data/placement/avail
ability-zone').Content
    $Region = $AZ.Substring(0, $AZ.Length -1)

    #Get the volumes attached to this instance
    $BlockDeviceMappings = (Get-EC2Instance -Region $Region -Instance $In
stanceId).Instances.BlockDeviceMappings
}
Catch
{
    Write-Host "Could not access the AWS API, therefore, VolumeId is not
available.
Verify that you provided your access keys."  -ForegroundColor Yellow
}

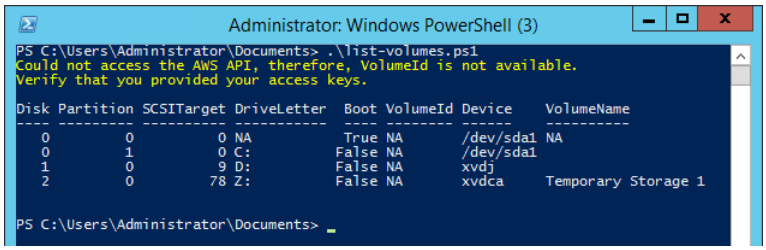
Get-WmiObject -Class Win32_DiskDrive | % {
    $Drive = $_
    # Find the partitions for this drive
    Get-WmiObject -Class Win32_DiskDriveToDiskPartition |  Where-Object
{$_.Antecedent -eq $Drive.Path.Path} | %{
        $D2P = $_
        # Get details about each partition
        $Partition = Get-WmiObject -Class Win32_DiskPartition |  Where-Object
{$_.Path.Path -eq $D2P.Dependent}
        # Find the drive that this partition is linked to
       $Disk = Get-WmiObject -Class Win32_LogicalDiskToPartition | Where-Object
 {$_.Antecedent -in $D2P.Dependent} | %{
            $L2P = $_
            #Get the drive letter for this partition, if there is one
           Get-WmiObject -Class Win32_LogicalDisk | Where-Object {$_.Path.Path
 -in $L2P.Dependent}
        }
       $BlockDeviceMapping = $BlockDeviceMappings | Where-Object {$_.DeviceName
 -eq $Map[$Drive.SCSITargetId.ToString()]}

        # Display the information in a table
        New-Object PSObject -Property @{
            Device = $Map[$Drive.SCSITargetId.ToString()];
            Disk = [Int]::Parse($Partition.Name.Split(",")[0].Replace("Disk
#",""));
            Boot = $Partition.BootPartition;
            Partition = [Int]::Parse($Partition.Name.Split(",")[1].Replace("
Partition #",""));
            SCSITarget = $Drive.SCSITargetId;
            DriveLetter = If($Disk -eq $NULL) {"NA"} else {$Disk.DeviceID};
            VolumeName = If($Disk -eq $NULL) {"NA"} else {$Disk.VolumeName};
            VolumeId = If($BlockDeviceMapping -eq $NULL) {"NA"} else {$Block
DeviceMapping.Ebs.VolumeId}
        }
    }
} | Sort-Object Disk, Partition | Format-Table -AutoSize -Property Disk, Parti
tion, SCSITarget, DriveLetter, Boot,
VolumeId, Device, VolumeName
```

Before you run this script, be sure to run the following command to enable PowerShell script execution.

```
Set-ExecutionPolicy RemoteSigned
```

Copy the script and save it as a `.ps1` file on the Windows instance. If you run the script without setting your access keys, you'll see output similar to the following.



If you specified an IAM role with a policy that allows access to Amazon EC2 when you launched the instance, or if you set up your credentials on the Windows instance as described in Using AWS Credentials in the *AWS Tools for Windows PowerShell User Guide*, you'll get the volume ID (vol-*xxxxxxxx*) for the EBS volumes in the VolumeId column instead of `NA`.

Disk Device to Device Name Mapping

The following table describes how the Citrix PV and AWS PV drivers map instance store volumes to Windows volumes. The number of available instance store volumes is determined by the instance type. For more information, see Instance Store Volumes (p. 578).

Location	Device Name
Bus Number 0, Target ID 78, LUN 0	xvdca
Bus Number 0, Target ID 79, LUN 0	xvdcb
Bus Number 0, Target ID 80, LUN 0	xvdcc
Bus Number 0, Target ID 81, LUN 0	xvdcd
Bus Number 0, Target ID 82, LUN 0	xvdce
Bus Number 0, Target ID 83, LUN 0	xvdcf
Bus Number 0, Target ID 84, LUN 0	xvdcg
Bus Number 0, Target ID 85, LUN 0	xvdch
Bus Number 0, Target ID 86, LUN 0	xvdci
Bus Number 0, Target ID 87, LUN 0	xvdcj
Bus Number 0, Target ID 88, LUN 0	xvdck
Bus Number 0, Target ID 89, LUN 0	xvdcl

The following table describes how the Citrix PV and AWS PV drivers map EBS volumes to Windows volumes. For more information, see Device Naming on Windows Instances (p. 586).

Location	Device Name
Bus Number 0, Target ID 0, LUN 0	/dev/sda1

Location	Device Name
Bus Number 0, Target ID 1, LUN 0	xvdb
Bus Number 0, Target ID 2, LUN 0	xvdc
Bus Number 0, Target ID 3, LUN 0	xvdd
Bus Number 0, Target ID 4, LUN 0	xvde
Bus Number 0, Target ID 5, LUN 0	xvdf
Bus Number 0, Target ID 6, LUN 0	xvdg
Bus Number 0, Target ID 7, LUN 0	xvdh
Bus Number 0, Target ID 8, LUN 0	xvdi
Bus Number 0, Target ID 9, LUN 0	xvdj
Bus Number 0, Target ID 10, LUN 0	xvdk
Bus Number 0, Target ID 11, LUN 0	xvdl
Bus Number 0, Target ID 12, LUN 0	xvdm
Bus Number 0, Target ID 13, LUN 0	xvdn
Bus Number 0, Target ID 14, LUN 0	xvdo
Bus Number 0, Target ID 15, LUN 0	xvdp
Bus Number 0, Target ID 16, LUN 0	xvdq
Bus Number 0, Target ID 17, LUN 0	xvdr
Bus Number 0, Target ID 18, LUN 0	xvds
Bus Number 0, Target ID 19, LUN 0	xvdt
Bus Number 0, Target ID 20, LUN 0	xvdu
Bus Number 0, Target ID 21, LUN 0	xvdv
Bus Number 0, Target ID 22, LUN 0	xvdw
Bus Number 0, Target ID 23, LUN 0	xvdx
Bus Number 0, Target ID 24, LUN 0	xvdy
Bus Number 0, Target ID 25, LUN 0	xvdz

Using Public Data Sets

Amazon Web Services provides a repository of public data sets that can be seamlessly integrated into AWS cloud-based applications. Amazon stores the data sets at no charge to the community and, as with all AWS services, you pay only for the compute and storage you use for your own applications.

Contents

- Public Data Set Concepts (p. 602)
- Finding Public Data Sets (p. 602)
- Creating a Public Data Set Volume from a Snapshot (p. 603)
- Attaching and Mounting the Public Data Set Volume (p. 604)

Public Data Set Concepts

Previously, large data sets such as the mapping of the Human Genome and the US Census data required hours or days to locate, download, customize, and analyze. Now, anyone can access these data sets from an EC2 instance and start computing on the data within minutes. You can also leverage the entire AWS ecosystem and easily collaborate with other AWS users. For example, you can produce or use prebuilt server images with tools and applications to analyze the data sets. By hosting this important and useful data with cost-efficient services such as Amazon EC2, AWS hopes to provide researchers across a variety of disciplines and industries with tools to enable more innovation, more quickly.

For more information, go to the Public Data Sets on AWS Page.

Available Public Data Sets

Public data sets are currently available in the following categories:

- **Biology—**Includes Human Genome Project, GenBank, and other content.
- **Chemistry—**Includes multiple versions of PubChem and other content.
- **Economics—**Includes census data, labor statistics, transportation statistics, and other content.
- **Encyclopedic—**Includes Wikipedia content from multiple sources and other content.

Finding Public Data Sets

Before you can use a public data set, you must locate the data set and determine which format the data set is hosted in. The data sets are available in two possible formats: Amazon EBS snapshots or Amazon S3 buckets.

To find a public data set and determine its format

1. Go to the Public Data Sets Page to see a listing of all available public data sets. You can also enter a search phrase on this page to query the available public data set listings.
2. Click the name of a data set to see its detail page.
3. On the data set detail page, look for a snapshot ID listing to identify an Amazon EBS formatted data set or an Amazon S3 URL.

Data sets that are in snapshot format are used to create new EBS volumes that you attach to an EC2 instance. For more information, see Creating a Public Data Set Volume from a Snapshot (p. 603).

For data sets that are in Amazon S3 format, you can use the AWS SDKs or the HTTP query API to access the information, or you can use the AWS CLI to copy or synchronize the data to and from your instance. For more information, see Amazon S3 and Amazon EC2 (p. 583).

You can also use Amazon Elastic MapReduce to analyze and work with public data sets. For more information, see What is Amazon EMR?.

Creating a Public Data Set Volume from a Snapshot

To use a public data set that is in snapshot format, you create a new volume, specifying the snapshot ID of the public data set. You can create your new volume using the AWS Management Console as follows. If you prefer, you can use the ec2-create-volume command instead.

To create a public data set volume from a snapshot

1. Open the Amazon EC2 console.
2. From the navigation bar, select the region that your data set snapshot is located in.

   Important
   Snapshot IDs are constrained to a single region, and you cannot create a volume from a snapshot that is located in another region. In addition, you can only attach an EBS volume to an instance in the same Availability Zone. For more information, see Resource Locations (p. 605).

   If you need to create this volume in a different region, you can copy the snapshot to your required region and then restore it to a volume in that region. For more information, see Copying an Amazon EBS Snapshot (p. 552).
3. In the navigation pane, click **Volumes**.
4. Above the upper pane, click **Create Volume**.
5. In the **Create Volume** dialog box, in the **Type** list, select **General Purpose (SSD)**, **Provisioned IOPS (SSD)**, or Magnetic. For more information, see Amazon EBS Volume Types (p. 520).
6. In the **Snapshot** field, start typing the ID or description of the snapshot for your data set. Select the snapshot from the list of suggested options.

   Note
   If the snapshot ID you are expecting to see does not appear, you may have a different region selected in the Amazon EC2 console. If the data set you identified in Finding Public Data Sets (p. 602) does not specify a region on its detail page, it is likely contained in the us-east-1 (N. Virginia) region.
7. In the **Size** field, enter the size of the volume (in GiB or TiB), or verify the that the default size of the snapshot is adequate.

   Note
   If you specify both a volume size and a snapshot ID, the size must be equal to or greater than the snapshot size. When you select a volume type and a snapshot ID, minimum and maximum sizes for the volume are shown next to the **Size** list.
8. For Provisioned IOPS (SSD) volumes, in the **IOPS** field, enter the maximum number of input/output operations per second (IOPS) that the volume can support.
9. In the **Availability Zone** list, select the Availability Zone in which to launch the instance.

   Important
   EBS volumes can only be attached to instances in the same Availability Zone.
10. Click **Yes, Create**.

Important
If you created a larger volume than the default size for that snapshot (by specifying a size in Step 7 (p. 603)), you need to extend the file system on the volume to take advantage of the extra space. For more information, see Expanding the Storage Space of an EBS Volume on Windows (p. 544).

Attaching and Mounting the Public Data Set Volume

After you have created your new data set volume, you need to attach it to an EC2 instance to access the data (this instance must also be in the same Availability Zone as the new volume). For more information, see Attaching an Amazon EBS Volume to an Instance (p. 529).

After you have attached the volume to an instance, you need to mount the volume on the instance. For more information, see Making an Amazon EBS Volume Available for Use (p. 530).

Resources and Tags

Amazon EC2 enables you to manage your Amazon EC2 resources, such as images, instances, volumes, and snapshots. For more information, see the following documentation.

Topics

- Resource Locations (p. 605)
- Listing and Filtering Your Resources (p. 606)
- Tagging Your Amazon EC2 Resources (p. 609)
- Amazon EC2 Service Limits (p. 618)
- Amazon EC2 Usage Reports (p. 619)

Resource Locations

The following table describes which Amazon EC2 resources are global, regional, or based on Availability Zone.

Resource	Type	Description
AWS Account	Global	You can use the same AWS account in all regions.
Key Pairs	Global or Regional	You can use the key pairs that you create using Amazon EC2 only in the region where you created them. You can create and upload an RSA key pair that you can use in all regions. For more information, see Amazon EC2 Key Pairs and Windows Instances (p. 394).
Amazon EC2 Resource Identifiers	Regional	Each resource identifier, such as an AMI ID, instance ID, EBS volume ID, or EBS snapshot ID, is tied to its region and can be used only in the region where you created the resource.

Resource	Type	Description
User-Supplied Resource Names	Regional	Each resource name, such as a security group name or key pair name, is tied to its region and can be used only in the region where you created the resource. Although you can create resources with the same name in multiple regions, they aren't related to each other.
AMIs	Regional	An AMI is tied to the region where its files are located within Amazon S3. You can copy an AMI from one region to another. For more information, see Copying an AMI (p. 74).
Elastic IP Addresses	Regional	An Elastic IP address is tied to a region and can be associated only with an instance in the same region.
Security Groups	Regional	A security group is tied to a region and can be assigned only to instances in the same region. You can't enable an instance to communicate with an instance outside its region using security group rules. Traffic from an instance in another region is seen as WAN bandwidth.
EBS Snapshots	Regional	An EBS snapshot is tied to its region and can only be used to create volumes in the same region. You can copy a snapshot from one region to another. For more information, see Copying an Amazon EBS Snapshot (p. 552).
EBS Volumes	Availability Zone	An Amazon EBS volume is tied to its Availability Zone and can be attached only to instances in the same Availability Zone.
Instances	Availability Zone	An instance is tied to the Availability Zones in which you launched it. However, note that its instance ID is tied to the region.

Listing and Filtering Your Resources

Amazon EC2 provides different *resources* that you can use. These resources include images, instances, volumes, and snapshots. You can get a list of some types of resource using the Amazon EC2 console. You can get a list of each type of resource using its corresponding command or API action. If you have many resources, you can filter the results to include only the resources that match certain criteria.

Topics

- Advanced Search (p. 606)
- Listing Resources Using the Console (p. 607)
- Filtering Resources Using the Console (p. 608)
- Listing and Filtering Using the CLI and API (p. 609)

Advanced Search

Advanced search allows you to search using a combination of filters to achieve precise results. You can filter by keywords, user-defined tag keys, and predefined resource attributes.

The specific search types available are:

- **Search by keyword**

  To search by keyword, type or paste what you're looking for in the search box, and then press Enter. For example, to search for a specific instance, you can type the instance ID.
- **Search by fields**

  You can also search by fields, tags, and attributes associated with a resource. For example, to find all instances in the stopped state:

  1. In the search box, start typing **`Instance State`**. As you type, you'll see a list of suggested fields.
  2. Select **Instance State** from the list.
  3. Select **Stopped** from the list of suggested values.
  4. To further refine your list, click the search box for more search options.
- **Advanced search**

  You can create advanced queries by adding multiple filters. For example, you can search by tags and see instances for the Flying Mountain project running in the Production stack, and then search by attributes to see all t2.micro instances, or all instances in us-west-2a, or both.
- **Inverse search**

  You can search for resources that do not match a specified value. For example, to list all instances that are not terminated, search by the **Instance State** field, and prefix the Terminated value with an exclamation mark (!).
- **Partial search**

  When searching by field, you can also enter a partial string to find all resources that contain the string in that field. For example, search by **Instance Type**, and then type **`t2`** to find all t2.micro, t2.small or t2.medium instances.
- **Regular expression**

  Regular expressions are useful when you need to match the values in a field with a specific pattern. For example, search by the Name tag, and then type **`^s.*`** to see all instances with a Name tag that start with an 's'.

After you have the precise results of your search, you can bookmark the URL for easy reference. In situations where you have thousands of instances, filters and bookmarks can save you a great deal of time; you don't have to run searches repeatedly.

Combining Search Filters

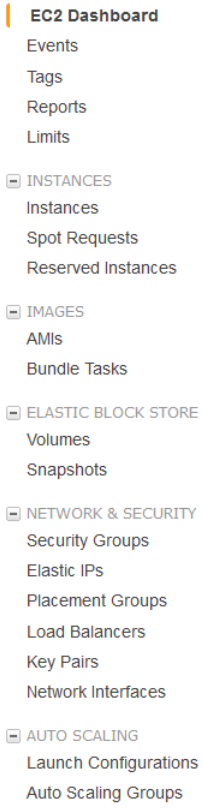
In general, multiple filters with the same key field (e.g., tag:Name, search, Instance State) are automatically joined with OR. This is intentional, as the vast majority of filters would not be logical if they were joined with AND. For example, you would get zero results for a search on Instance State=running AND Instance State=stopped. In many cases, you can granulate the results by using complementary search terms on different key fields, where the AND rule is automatically applied instead. If you search for tag: Name:=All values and tag:Instance State=running, you get search results that contain both those criteria. To fine-tune your results, simply remove one filter in the string until the results fit your requirements.

Listing Resources Using the Console

You can view the most common Amazon EC2 resource types using the console. To view additional resources, use the command line interface or the API actions.

To list EC2 resources using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click the option that corresponds to the resource, such as **AMIs** or **Instances**.



3. The page displays all the available resources.

Filtering Resources Using the Console

You can perform filtering and sorting of the most common resource types using the Amazon EC2 console. For example, you can use the search bar on the instances page to sort instances by tags, attributes, or keywords.

You can also use the search field on each page to find resources with specific attributes or values. You can use regular expressions to search on partial or multiple strings. For example, to find all instances that are using the MySG security group, enter `MySG` in the search field. The results will include any values that contain `MySG` as a part of the string, such as `MySG2` and `MySG3`. To limit your results to MySG only, enter `\bMySG\b` in the search field. To list all the instances whose type is either `m1.small` or `m1.large`, enter `m1.small|m1.large` in the search field.

To list volumes in the `us-east-1b` Availability Zone with a status of `available`

1. In the navigation pane, click **Volumes**.
2. Click on the search box, select **Attachment Status** from the menu, and then select **Detached**. (A detached volume is available to be attached to an instance in the same Availability Zone.)
3. Click on the search box again, select **State**, and then select **Available**.
4. Click on the search box again, select **Availability Zone**, and then select `us-east-1b`.

5. Any volumes that meet this criteria are displayed.

To list public 64-bit Windows AMIs backed by Amazon EBS

1. In the navigation pane, click **AMIs**.
2. In the **Filter** pane, select **Public images**, **EBS images**, and then **Windows** from the **Filter** lists.
3. Enter `x86_64` in the search field.
4. Any AMIs that meet this criteria are displayed.

Listing and Filtering Using the CLI and API

Each resource type has a corresponding CLI command or API request that you use to list resources of that type. For example, you can list Amazon Machine Images (AMI) using `ec2-describe-images` or `DescribeImages`. The response contains information for all your resources.

The resulting lists of resources can be long, so you might want to filter the results to include only the resources that match certain criteria. You can specify multiple filter values, and you can also specify multiple filters. For example, you can list all the instances whose type is either `m1.small` or `m1.large`, and that have an attached EBS volume that is set to delete when the instance terminates. The instance must match all your filters to be included in the results.

> **Note**
> If you use a tag filter, the response includes the tags for your resources; otherwise, tags may be omitted in the response.

You can also use wildcards with the filter values. An asterisk (*) matches zero or more characters, and a question mark (?) matches exactly one character. For example, you can use `*database*` as a filter value to get all EBS snapshots that include `database` in the description. If you were to specify `database` as the filter value, then only snapshots whose description equals `database` would be returned. Filter values are case sensitive. We support only exact string matching, or substring matching (with wildcards). If a resulting list of resources is long, using an exact string filter may return the response faster.

> **Tip**
> Your search can include the literal values of the wildcard characters; you just need to escape them with a backslash before the character. For example, a value of `\*amazon\?\\` searches for the literal string `*amazon?\`.

For a list of supported filters per Amazon EC2 resource, see the relevant documentation:

- For the AWS CLI, see the relevant `describe` command in the *AWS Command Line Interface Reference*.
- For the Amazon EC2 CLI, see the relevant `ec2-describe` command in the *Amazon EC2 Command Line Reference*.
- For Windows PowerShell, see the relevant `Get` command in the *AWS Tools for Windows PowerShell Reference*.
- For the Query API, see the relevant `Describe` API action in the *Amazon EC2 API Reference*.

Tagging Your Amazon EC2 Resources

To help you manage your instances, images, and other Amazon EC2 resources, you can assign your own metadata to each resource in the form of *tags*. This topic describes tags and shows you how to create them.

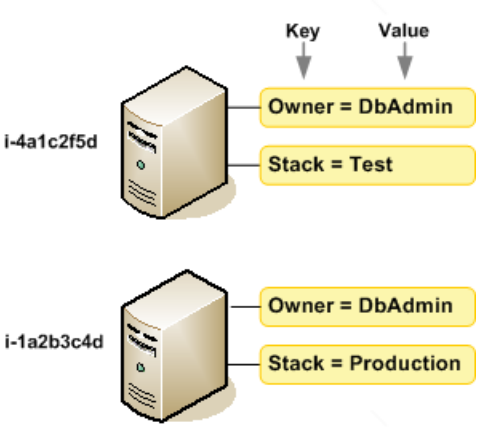
Contents

- Tag Basics (p. 610)
- Tag Restrictions (p. 610)
- Tagging Your Resources for Billing (p. 612)
- Working with Tags Using the Console (p. 612)
- Working with Tags Using the CLI or API (p. 617)

Tag Basics

Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. Each tag consists of a key and an optional value, both of which you define. For example, you could define a set of tags for your account's Amazon EC2 instances that helps you track each instance's owner and stack level. We recommend that you devise a set of tag keys that meets your needs for each resource type. Using a consistent set of tag keys makes it easier for you to manage your resources. You can search and filter the resources based on the tags you add.

The following diagram illustrates how tagging works. In this example, you've assigned two tags to each of your instances, one called `Owner` and another called `Stack`. Each of the tags also has an associated value.



Tags don't have any semantic meaning to Amazon EC2 and are interpreted strictly as a string of characters. Also, tags are not automatically assigned to your resources.

You can work with tags using the AWS Management Console, the Amazon EC2 command line interface (CLI), and the Amazon EC2 API.

You can assign tags only to resources that already exist. When you use the Amazon EC2 console, you can access a list of tags to add to an instance, which will be applied immediately after the instance is created. If you add a tag that has the same key as an existing tag on that resource, the new value overwrites the old value. You can edit tag keys and values, and you can remove tags from a resource at any time. You can set a tag's value to the empty string, but you can't set a tag's value to null.

If you're using AWS Identity and Access Management (IAM), you can control which users in your AWS account have permission to create, edit, or delete tags. For more information about IAM, see Controlling Access to Amazon EC2 Resources (p. 406).

Tag Restrictions

The following basic restrictions apply to tags:

- Maximum number of tags per resource—10
- Maximum key length—127 Unicode characters in UTF-8
- Maximum value length—255 Unicode characters in UTF-8
- Tag keys and values are case sensitive.
- Do not use the `aws:` prefix in your tag names or values because it is reserved for AWS use. You can't edit or delete tag names or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

> **Note**
> If you plan to use a tagging schema across multiple services and resources, keep in mind that while there are no restrictions on special characters for Amazon EC2, other services may have different restrictions or limits. Generally allowed characters are: letters, spaces, and numbers representable in UTF-8, plus the following special characters: + - = . _ : / @.

You can't terminate, stop, or delete a resource based solely on its tags; you must specify the resource identifier. For example, to delete snapshots that you tagged with a tag key called `DeleteMe`, you must use the `DeleteSnapshots` action with the resource identifiers of the snapshots, such as `snap-1a2b3c4d`. To identify resources by their tags, you can use the `DescribeTags` action to list all of your tags and their associated resources. You can also filter by resource type or tag keys and values. You can't call `DeleteSnapshots` with a filter that specified the tag. For more information about using filters when listing your resources, see Listing and Filtering Your Resources (p. 606).

You can tag public or shared resources, but the tags you assign are available only to your AWS account and not to the other accounts sharing the resource.

You can't tag all resources, and some you can only tag using API actions or the command line. The following table lists all Amazon EC2 resources and the tagging restrictions that apply to them, if any. Resources with tagging restrictions of None can be tagged with API actions, the CLI, and the console.

Resource	Tagging support	Tagging restrictions
AMI	Yes	None
Bundle task	No	
Customer gateway	Yes	None
DHCP option	Yes	None
EBS volume	Yes	None
Instance store volume	No	
Elastic IP	No	
Instance	Yes	None
Internet gateway	Yes	None
Key pair	No	
Network ACL	Yes	None
Network interface	Yes	None
Placement group	No	
Reserved Instance	Yes	None

Resource	Tagging support	Tagging restrictions
Reserved Instance Listing	No	
Route table	Yes	None
Spot instance request	Yes	None
Security group - EC2 Classic	Yes	None
Security group - VPC	Yes	None
Snapshot	Yes	None
Subnet	Yes	None
Virtual private gateway	Yes	None
VPC	Yes	None
VPC endpoint	No	
VPC flow log	No	
VPC peering connection	Yes	None
VPN connection	Yes	None

For more information about tagging using the AWS Management Console, see Working with Tags Using the Console (p. 612). For more information about tagging using the API or command line, see Working with Tags Using the CLI or API (p. 617).

Tagging Your Resources for Billing

You can use tags to organize your AWS bill to reflect your own cost structure. To do this, sign up to get your AWS account bill with tag key values included. For more information about setting up a cost allocation report with tags, see Setting Up Your Monthly Cost Allocation Report in *About AWS Account Billing*. To see the cost of your combined resources, you can organize your billing information based on resources that have the same tag key values. For example, you can tag several resources with a specific application name, and then organize your billing information to see the total cost of that application across several services. For more information, see Cost Allocation and Tagging in *About AWS Account Billing*.

Note
If you've just enabled reporting, the current month's data will be available for viewing in about 24 hours.

Working with Tags Using the Console

Using the Amazon EC2 console, you can see which tags are in use across all of your Amazon EC2 resources in the same region. You can view tags by resource and by resource type, and you can also view how many items of each resource type are associated with a specified tag. You can also use the Amazon EC2 console to apply or remove tags from one or more resources at a time.

For ease of use and best results, use Tag Editor in the AWS Management Console, which provides a central, unified way to create and manage your tags. For more information, see Working with Tag Editor in Getting Started with the AWS Management Console.

Contents

- Displaying Tags (p. 613)

- Adding and Deleting Tags on an Individual Resource (p. 613)
- Adding and Deleting Tags to a Group of Resources (p. 614)
- Adding a Tag When You Launch an Instance (p. 616)
- Filtering a List of Resources by Tag (p. 616)

Displaying Tags

You can display tags in two different ways in the Amazon EC2 console. You can display the tags for an individual resource or for all resources.

To display tags for individual resources

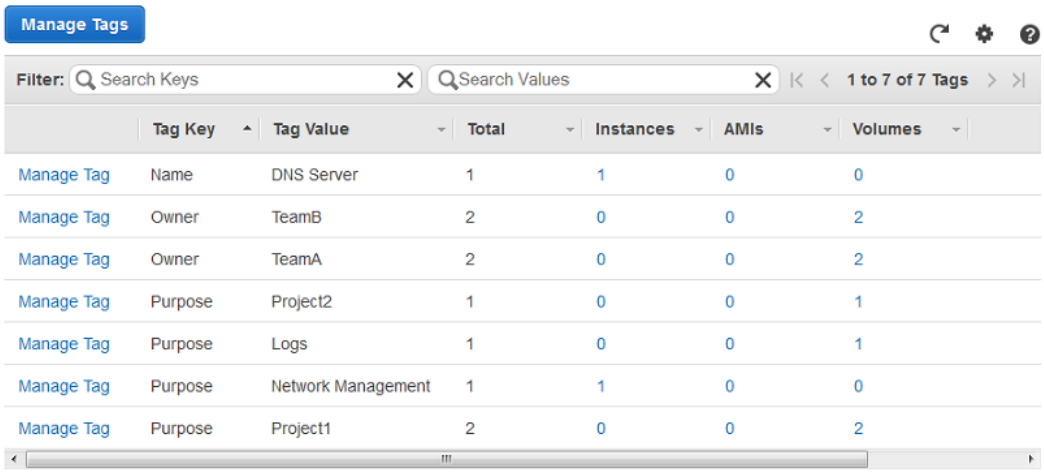
When you select a resource-specific page in the Amazon EC2 console, it displays a list of those resources. For example, if you select **Instances** from the navigation pane, the console displays a list of Amazon EC2 instances. When you select a resource from one of these lists (e.g., an instance), if the resource supports tags, you can view and manage its tags. On most resource pages, you can view the tags in the **Tags** tab on the details pane. The following image shows the **Tags** tab for an instance with two tags: Name = DNS Server and Purpose = Network Management.

You can add a column to the resource list that displays all values for tags with the same key. This column enables you to sort and filter the resource list by the tag. There are two ways to add a new column to the resource list to display your tags.

- On the **Tags** tab, click **Show Column** for the tag.
- Click the **Show/Hide Columns** gear-shaped icon, and in the **Show/Hide Columns** dialog box, select the tag key under **Your Tag Keys**.

To display tags for all resources

You can display tags across all resources by selecting **Tags** from the navigation pane in the Amazon EC2 console. The following image shows the **Tags** pane, which lists all tags in use by resource type.

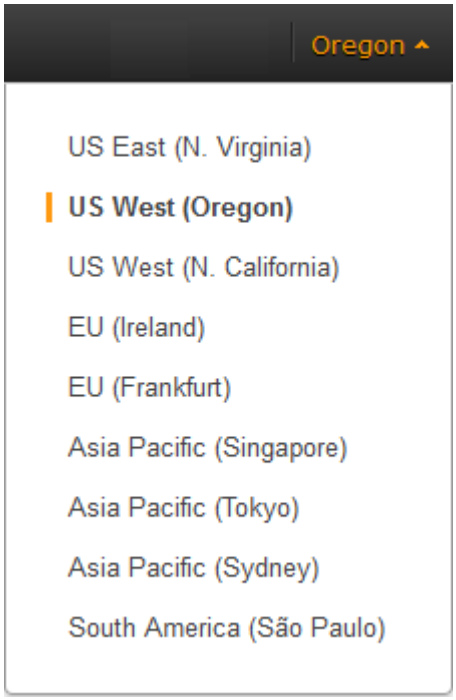


Adding and Deleting Tags on an Individual Resource

You can manage tags for an individual resource directly from the resource's page. If you are managing an AMI's tags, the procedures are different from that of other resources. All procedures are explained below.

To add a tag to an individual resource

1. Open the Amazon EC2 console.
2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see Resource Locations (p. 605).



3. In the navigation pane, click a resource type (for example, **Instances**).
4. Select the resource from the resource list.
5. Select the **Tags** tab in the details pane.
6. Click the **Add/Edit Tags** button.
7. In the **Add/Edit Tags** dialog box, specify the key and value for each tag, and then click **Save**.

To delete a tag from an individual resource

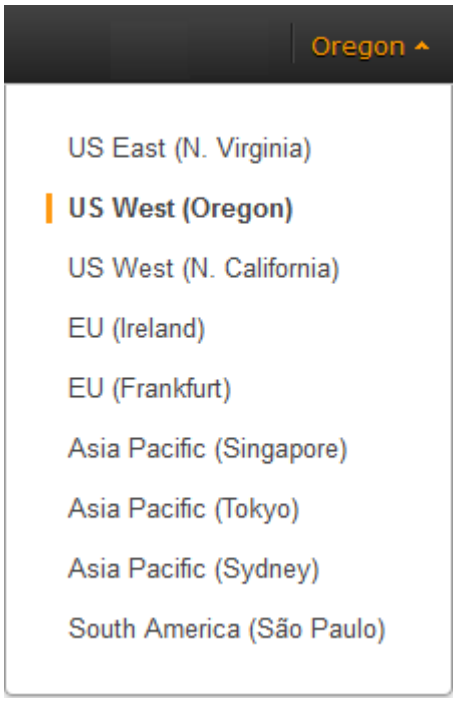
1. Open the Amazon EC2 console.
2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see Resource Locations (p. 605).
3. In the navigation pane, click a resource type (for example, **Instances**).
4. Select the resource from the resource list.
5. Select the **Tags** tab in the details pane.
6. Click **Add/Edit Tags**, click the **Delete** icon for the tag, and click **Save**.

Adding and Deleting Tags to a Group of Resources

To add a tag to a group of resources

1. Open the Amazon EC2 console.

2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see Resource Locations (p. 605).



3. In the navigation pane, click **Tags**.
4. At the top of the content pane, click **Manage Tags**.
5. From the **Filter** drop-down list, select the type of resource (for example, instances) that you want to add tags to.
6. In the resources list, select the check box next to each resource that you want to add tags to.
7. In the **Key** and **Value** boxes under **Add Tag**, type the tag key and values you want, and then click **Add Tag**.

   Note
   If you add a new tag with the same tag key as an existing tag, the new tag overwrites the existing tag.

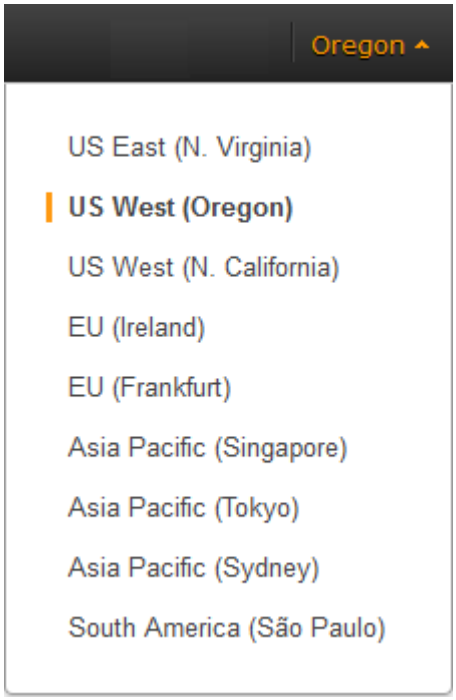
To remove a tag from a group of resources

1. Open the Amazon EC2 console.
2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see Resource Locations (p. 605).
3. In the navigation pane, click **Tags**.
4. At the top of the content pane, click **Manage Tags**.
5. To view the tags in use, click the **Show/Hide Columns** gear-shaped icon, and in the **Show/Hide Columns** dialog box, select the tag keys you want to view, and then click **Close**.
6. From the **Filter** drop-down list, select the type of resource (for example, instances) that you want to remove tags from.
7. In the resource list, select the check box next to each resource that you want to remove tags from.
8. Under **Remove Tag**, click in the **Key** box to select a key, or type its name, and then click **Remove Tag**.

Adding a Tag When You Launch an Instance

To add a tag using the Launch Wizard

1. From the navigation bar, select the region for the instance. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. Select the region that meets your needs. For more information, see Resource Locations (p. 605).



2. Click the **Launch Instance** button on the EC2 dashboard.
3. The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations called Amazon Machine Images (AMIs). Choose the AMI that you want to use and click its **Select** button. For more information about selecting an AMI, see Finding a Windows AMI (p. 56).
4. On the **Configure Instance Details** page, configure the instance settings as necessary, and then click **Next: Add Storage**.
5. On the **Add Storage** page, you can specify additional storage volumes for your instance. Click **Next: Tag Instance** when done.
6. On the **Tag Instance** page, specify tags for the instance by providing key and value combinations. Click **Create Tag** to add more than one tag to your instance. Click **Next: Configure Security Group** when you are done.
7. On the **Configure Security Group** page, you can choose from an existing security group that you own, or let the wizard create a new security group for you. Click **Review and Launch** when you are done.
8. Review your settings. When you're satisfied with your selections, click **Launch**. Select an existing key pair or create a new one, select the acknowledgment check box, and then click **Launch Instances**.

Filtering a List of Resources by Tag

You can filter your list of resources based on one or more tag keys and tag values.

To filter a list of resources by tag

1. Display a column for the tag as follows:
   a. Select one of the resources.
   b. Select the **Tags** tab in the details pane.
   c. Locate the tag in the list and click **Show Column**.
2. Click the filter icon in the top right corner of the column for the tag to display the filter list.
3. Select the tag values, and then click **Apply Filter** to filter the results list.

   Note
   For more information about filters see Listing and Filtering Your Resources (p. 606).

Working with Tags Using the CLI or API

Use the following to add, update, list, and delete the tags for your resources. The corresponding documentation provides examples.

Task	AWS CLI	Amazon EC2 CLI	AWS Tools for Windows PowerShell	API Action
Add or overwrite one or more tags.	create-tags	ec2-create-tags	New-EC2Tag	CreateTags
Delete one or more tags.	delete-tags	ec2-delete-tags	Remove-EC2Tag	DeleteTags
Describe one or more tags.	describe-tags	ec2-describe-tags	Get-EC2Tag	DescribeTags

You can also filter a list of resources according to their tags. The following examples demonstrate how to filter your instances using tags with the describe-instances command.

Example 1: Describe instances with the specified tag key

The following command describes the instances with a Stack tag, regardless of the value of the tag.

```
aws ec2 describe-instances --filters Name=tag-key,Values=Stack
```

Example 2: Describe instances with the specified tag

The following command describes the instances with the tag Stack=production.

```
aws ec2 describe-instances --filters Name=tag:Stack,Values=production
```

Example 3: Describe instances with the specified tag value

The following command describes the instances with a tag with the value production, regardless of the tag key.

```
aws ec2 describe-instances --filters Name=tag-value,Values=production
```

Amazon EC2 Service Limits

Amazon EC2 provides different *resources* that you can use. These resources include images, instances, volumes, and snapshots. When you create your AWS account, we set default limits on these resources on a per-region basis. For example, there is a limit on the number of instances that you can launch in a region. Therefore, when you launch an instance in the US West (Oregon) region, the request must not cause your usage to exceed your current instance limit in that region.

The Amazon EC2 console provides limit information for the resources managed by the Amazon EC2 and Amazon VPC consoles. You can request an increase for many of these limits. Use the limit information that we provide to manage your AWS infrastructure. Plan to request any limit increases in advance of the time that you'll need them.

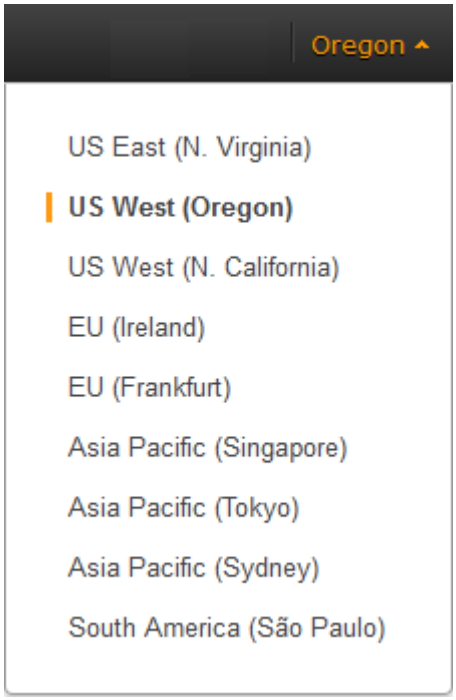
For more information about the limits for other services, see AWS Service Limits in the *Amazon Web Services General Reference*.

Viewing Your Current Limits

Use the **EC2 Service Limits** page in the Amazon EC2 console to view the current limits for resources provided by Amazon EC2 and Amazon VPC, on a per-region basis.

To view your current limits

1. Open the Amazon EC2 console.
2. From the navigation bar, select a region.



3. From the navigation pane, click **Limits**.
4. Locate the resource in the list. The **Current Limit** column displays the current maximum for that resource for your account.

Requesting a Limit Increase

Use the **Limits** page in the Amazon EC2 console to request an increase in the limits for resources provided by Amazon EC2 or Amazon VPC, on a per-region basis.

To request a limit increase

1. Open the Amazon EC2 console.
2. From the navigation bar, select a region.
3. From the navigation pane, click **Limits**.
4. Locate the resource in the list. Click **Request limit increase**.
5. Complete the required fields on the limit increase form. We'll respond to you using the contact method that you specified.

Amazon EC2 Usage Reports

The usage reports provided by Amazon EC2 enable you to analyze the usage of your instances in depth. The data in the usage reports is updated multiple times each day. You can filter the reports by AWS account, region, Availability Zone, operating system, instance type, purchasing option, tenancy, and tags.

To get usage and cost data for an account, you must have its account credentials and enable detailed billing reports with resources and tags for the account. If you're using consolidated billing and are logged into the payer account, you can view data for the payer account and all its linked accounts. If you're using consolidated billing and are logged into one of the linked accounts, you can only view data for that linked account. For information about consolidated billing, see Pay Bills for Multiple Accounts with Consolidated Billing.

Topics

- Available Reports (p. 619)
- Getting Set Up for Usage Reports (p. 619)
- Granting IAM Users Access to the Amazon EC2 Usage Reports (p. 621)
- Instance Usage Report (p. 621)
- Reserved Instance Utilization Reports (p. 625)

Available Reports

You can generate the following reports:

- Instance usage report (p. 621). This report covers your usage of On-Demand instances, Spot instances, and Reserved Instances.
- Reserved Instances utilization report (p. 625). This report covers the usage of your capacity reservation.

Getting Set Up for Usage Reports

Before you begin, enable detailed billing reports with resources and tags as shown in the following procedure. After you complete this procedure, we'll start collecting usage data for your instances. If you've already enabled detailed billing reports, you can access the usage data that we've been collecting since you enabled them.

Important
To complete these procedures, you must log in using your AWS account credentials. You can't complete these procedures if you log in using IAM user credentials.

To enable detailed billing reports

1. Select an existing Amazon S3 bucket to receive your usage data. Be sure to manage access to this bucket as it contains your billing data. (We don't require that you keep these files; in fact, you can delete them immediately if you don't need them.) If you don't have a bucket, create one as follows:
   a. Open the Amazon S3 console.
   b. Click **Create Bucket**.
   c. In the **Create a Bucket** dialog box, enter a name for your bucket (for example, *username*-ec2-usage-data), select a region, and then click **Create**. For more information about the requirements for bucket names, see Creating a Bucket in the *Amazon Simple Storage Service Console User Guide*.
2. Open the Billing and Cost Management console at https://console.aws.amazon.com/billing/home?#.
3. Click **Preferences** in the navigation pane.
4. Select **Receive Billing Reports**.
5. Specify the name of your Amazon S3 bucket in **Save to S3 Bucket**, and then click **Verify**.
6. Grant AWS permission to publish usage data to your Amazon S3 bucket.
   a. Under **Receive Billing Reports**, click **sample policy**. Copy the sample policy. Notice that the sample policy uses the bucket name you specified.
   b. Open the Amazon S3 console in another browser tab. Select your bucket, click **Properties**, and then expand **Permissions**. In the **Permissions** section, click **Add bucket policy**. Paste the sample policy into the text area and click **Save**. In the **Permissions** section, click **Save**.
   c. Return to the browser tab with the sample policy and click **Done**.
7. Under **Report**, select **Detailed billing report with resources and tags**.
8. Click **Save preferences**.

   Note
   It can take up to a day before you can see your data in the reports.

You can categorize your instances using tags. After you tag your instances, you must enable reporting on these tags.

To enable usage reporting by tag

1. Tag your instances. For best results, ensure that you add each tag you plan to use for reporting to each of your instances. For more information about how to tag an instance, see Tagging Your Amazon EC2 Resources (p. 609).
2. Open the Billing and Cost Management console at https://console.aws.amazon.com/billing/home?#.
3. Click **Preferences** in the navigation pane.
4. Under **Report**, click **Manage report tags**.
5. The page displays the list of tags that you've created. Select the tags that you'd like to use to filter or group your instance usage data, and then click **Save**. We automatically exclude any tags that you don't select from your instance usage report.

Note
We apply these changes only to the data for the current month. It can take up to a day for these changes to take effect.

Granting IAM Users Access to the Amazon EC2 Usage Reports

By default, IAM users can't access the Amazon EC2 usage reports. You must create an IAM policy that grants IAM users permission to access these reports.

The following policy allows users to view both Amazon EC2 usage reports.

```
{
  "Version": "2012-10-17",
  "Statement":[{
    "Effect": "Allow",
    "Action": "ec2-reports:*",
    "Resource": "*"
  }
  ]
}
```

The following policy allows users to view the instance usage report.

```
{
  "Version": "2012-10-17",
  "Statement":[{
    "Effect": "Allow",
    "Action": "ec2-reports:ViewInstanceUsageReport",
    "Resource": "*"
  }
  ]
}
```

The following policy allows users to view the Reserved Instances utilization report.

```
{
  "Version": "2012-10-17",
  "Statement":[{
    "Effect": "Allow",
    "Action": "ec2-reports:ViewReservedInstanceUtilizationReport",
    "Resource": "*"
  }
  ]
}
```

For more information, see Permissions and Policies in the *IAM User Guide*.

Instance Usage Report

You can use the instance usage report to view your instance usage and cost trends. You can see your usage data in either instance hours or cost. You can choose to see hourly, daily and monthly aggregates

of your usage data. You can filter or group the report by region, Availability Zone, instance type, AWS account, platform, tenancy, purchase option, or tag. After you configure a report, you can bookmark it so that it's easy to get back to later.

Here's an example of some of the questions that you can answer by creating an instance usage report:

- How much am I spending on instances of each instance type?
- How many instance hours are being used by a particular department?
- How is my instance usage distributed across Availability Zones?
- How is my instance usage distributed across AWS accounts?

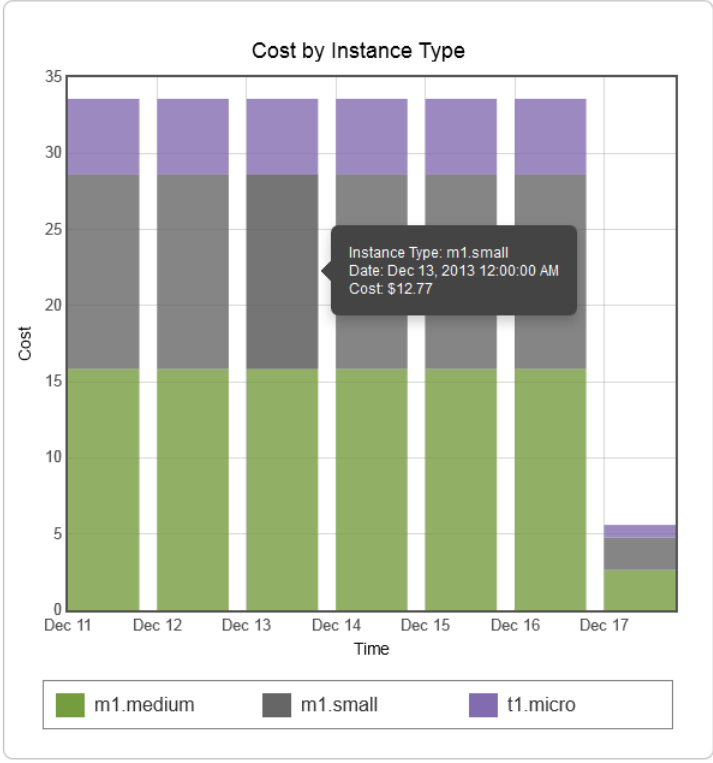
Topics

- Report Formats (p. 622)
- Viewing Your Instance Usage (p. 623)
- Bookmarking a Customized Report (p. 624)
- Exporting Your Usage Data (p. 624)

Report Formats

We display the usage data that you request as both a graph and a table.

For example, the following graph displays cost by instance type. The key for the graph indicates which color represents which instance type. To get detailed information about a segment of a bar, hover over it.



The corresponding table displays one column for each instance type. Notice that we include a color band in the column head that is the same color as the instance type in the graph.

Time (UTC)	m1.medium	m1.small	t1.micro
12/11/13	$15.84	$12.77	$4.97
12/12/13	$15.84	$12.77	$4.97
12/13/13	$15.84	$12.77	$4.97
12/14/13	$15.84	$12.77	$4.97
12/15/13	$15.84	$12.77	$4.97
12/16/13	$15.84	$12.77	$4.97
12/17/13	$2.64	$2.13	$0.83
Total	$97.68	$78.75	$30.65

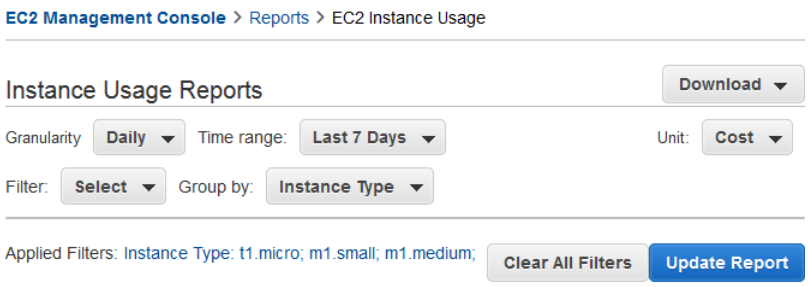
Viewing Your Instance Usage

The following procedures demonstrate how to generate usage reports using some of the capabilities we provide.

Before you begin, you must get set up. For more information, see Getting Set Up for Usage Reports (p. 619).

To filter and group your instance usage by instance type

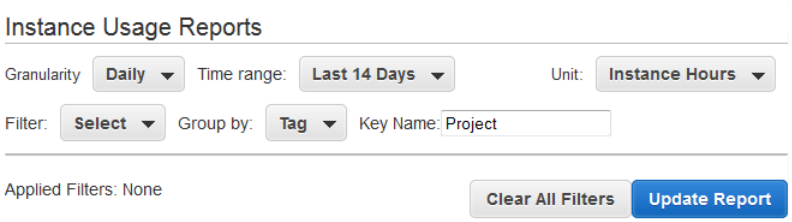
1. Open the Amazon EC2 console.
2. In the navigation pane, click **Reports** and then click **EC2 Instance Usage Report**.
3. Select an option for **Unit**. To view the time that your instances have been running, in hours, select `Instance Hours`. To view the cost of your instance usage, select `Cost`.
4. Select options for **Granularity** and **Time range**.
   - To view the data summarized for each hour in the time range, select `Hourly` granularity. You can select a time range of up to 2 days when viewing hourly data.
   - To view the data summarized for each day in the time range, select `Daily` granularity. You can select a time range of up to 2 months when viewing daily data.
   - To view the data summarized for each month in the time range, select `Monthly` granularity.
5. In the **Filter** list, select `Instance Type`. In the **Group by** list, select `Instance Type`.
6. In the filter area, select one or more instance types and then click **Update Report**. The filters you specify appear under **Applied Filters**.



Notice that you can return to the Amazon EC2 console by clicking either **Reports** or **EC2 Management Console** at the top of the page.

To group your instance usage based on tags

1. Open the Instance Usage Reports page.
2. Select an option for **Unit**. To view the time that your instances have been running, in hours, select `Instance Hours`. To view the cost of your instance usage, select `Cost`.
3. Select options for **Granularity** and **Time range**.
   - To view the data summarized for each hour in the time range, select `Hourly` granularity. You can select a time range of up to 2 days when viewing hourly data.
   - To view the data summarized for each day in the time range, select `Daily` granularity. You can select a time range of up to 2 months when viewing daily data.
   - To view the data summarized for each month in the time range, select `Monthly` granularity.
4. In the **Group by** list, select **Tag**.
5. Click the **Key Name** box, select a name from the list, and then click **Update Report**. If there are no items in this list, you must enable usage reporting by tag. For more information, see To enable usage reporting by tag (p. 620).



Bookmarking a Customized Report

You might want to generate a customized report again. Do this by bookmarking the report.

To bookmark a custom report

1. Select the options and filters for your report. Each selection you make adds a parameter to the console URL. For example, `granularity=Hourly` and `Filters=`*filter_list*.
2. Using your browser, add the console URL as a bookmark.
3. To generate the same report in the future, use the bookmark that you created.

Exporting Your Usage Data

You might want to include your report graph or table in other reports. Do this by exporting the data.

To export usage data

1. Select the options and filters for your report.
2. To export the usage data from the table as a `.csv` file, click **Download** and select **CSV Only**.
3. To export the graphical usage data as a `.png` file, click **Download** and select **Graph Only**.

Reserved Instance Utilization Reports

The Reserved Instance utilization report describes the utilization over time of each group (or *bucket*) of Amazon EC2 Reserved Instances that you own. Each bucket has a unique combination of region, Availability Zone, instance type, tenancy, offering type, and platform. You can specify the time range that the report covers, from a custom range to weeks, months, a year, or three years. The available data depends on when you enable detailed billing reports for the account (see Getting Set Up for Usage Reports (p. 619)). The Reserved Instance utilization report compares the Reserved Instance prices paid for instance usage in the bucket with On-Demand prices and shows your savings for the time range covered by the report.

To get usage and cost data for an account, you must have its account credentials and enable detailed billing reports with resources and tags for the account. If you're using consolidated billing and are logged into the payer account, you can view data for the payer account and all its linked accounts. If you're using consolidated billing and are logged into one of the linked accounts, you can only view data for that linked account. For information about consolidated billing, see Pay Bills for Multiple Accounts with Consolidated Billing.

> **Note**
> The Reserved Instance buckets aggregate Reserved Instances across Amazon VPC and non-Amazon VPC (EC2 Classic) network platform types in the same way that your bill is calculated. Additionally, Reserved Instances in a bucket may have different upfront and hourly prices.

Here are examples of some of the questions that you can answer using the Reserved Instance utilization report:

- How well am I utilizing my Reserved Instances?
- Are my Reserved Instances helping me save money?

Before you begin, you must get set up. For more information, see Getting Set Up for Usage Reports (p. 619).

Topics

- Getting to Know the Report (p. 625)
- Viewing Your Reserved Instance Utilization (p. 627)
- Bookmarking a Customized Report (p. 627)
- Exporting Your Usage Data (p. 628)
- Options Reference (p. 628)

Getting to Know the Report

The Reserved Instance utilization report displays your requested utilization data in graph and table formats.

The report aggregates Reserved Instance usage data for a given period by bucket. In the report, each row in the table represents a bucket and provides the following metrics:

- **Count**—The highest number of Reserved Instances owned at the same time during the period of the report.
- **Usage Cost**—The total Reserved Instance usage fees applied to instance usage covered by the Reserved Instance bucket.
- **Total Cost**—The usage cost plus the amortized upfront fee for the usage period associated with the Reserved Instance bucket.

Note
If the bucket contains a Reserved Instance that you sold in the Reserved Instances Marketplace and that Reserved Instance was active at any point during the period of the report, the total cost of the bucket might be inflated and your savings might be underestimated.

- **Savings**—The difference between what your usage for the period would have cost at On-Demand prices and what it actually cost using Reserved Instances (Total Cost).
- **Average Utilization**—The average hourly utilization rate for the Reserved Instance bucket over the period.
- **Maximum Utilization**—The highest utilization rate of any hour during the period covered by the report.

For each row—or Reserved Instance bucket—in the table, the graph represents data based on your selected **Show** metric over the selected **Time range** for the report. Each point in the graph represents a metric at a point in time. For information about report options, see Options Reference (p. 628).

A color band at the edge of each selected row in the table corresponds to a report line in the graph. You can show a row in the graph by selecting the checkbox at the beginning of the row.

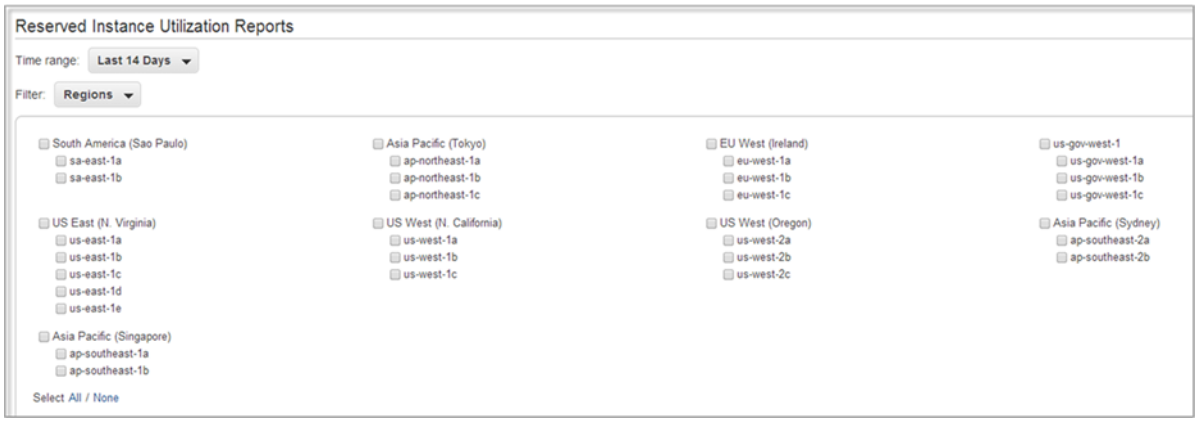
By default, the Reserved Instance utilization report returns data over the last 14 days for all Reserved Instance buckets. The graph shows the average utilization for the first five buckets in the table. You can customize the report graph to show different utilization (average utilization, maximum utilization) or cost (total cost, usage cost) data over a period ranging from 7 days to weeks, months, or years.

Customizing the Report

You can customize the Reserved Instance utilization report with **Time range** and **Filter** options.

Time range provides a list of common relative time ranges, ranging from **Last 7 Days** to **Last 3 Years**. Select the time range that works best for your needs, and then click **Update Report** to apply the change. To apply a time range that is not on the list, select **Custom** and enter the start date and end date for which you want to run the report.

Filter lets you scope your Reserved Instance utilization report by one or more of the following Reserved Instance qualities: region, instance type, accounts, platforms, tenancy, and offering types. For example, you can filter by region or by specific Availability Zones in a region, or both. To filter by region, select **Regions**, then select the regions and Availability Zones you want to include in the report, and click **Update Report**.

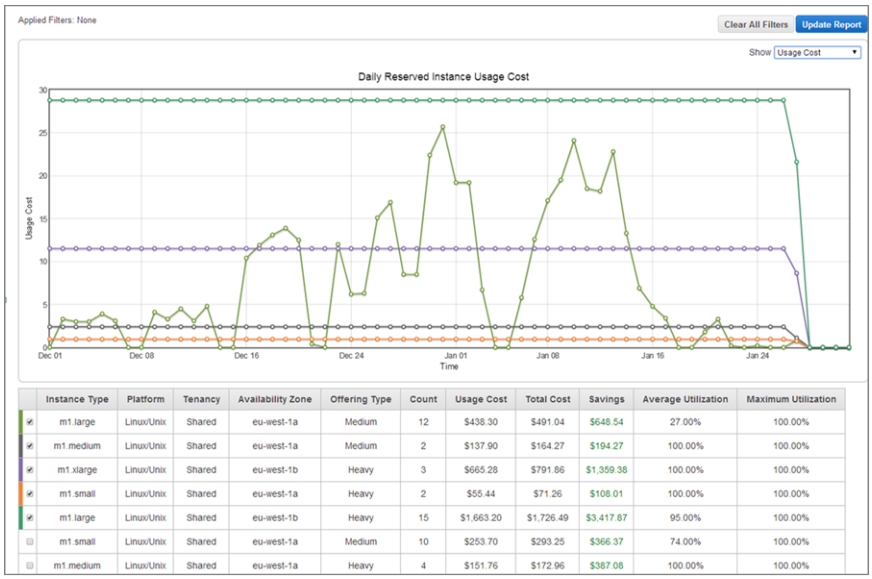


The report will return all results if no filter is applied.

For information about report options, see Options Reference (p. 628).

Viewing Your Reserved Instance Utilization

In this section, we will highlight aspects of your Reserved Instance utilization that the graph and table capture. For the purposes of this discussion, we'll use the following report, which is based on test data.



Instance Type	Platform	Tenancy	Availability Zone	Offering Type	Count	Usage Cost	Total Cost	Savings	Average Utilization	Maximum Utilization
m1.large	Linux/Unix	Shared	eu-west-1a	Medium	12	$438.30	$491.04	$648.54	27.00%	100.00%
m1.medium	Linux/Unix	Shared	eu-west-1a	Medium	2	$137.90	$164.27	$194.27	100.00%	100.00%
m1.xlarge	Linux/Unix	Shared	eu-west-1b	Heavy	3	$665.28	$791.86	$1,359.38	100.00%	100.00%
m1.small	Linux/Unix	Shared	eu-west-1a	Heavy	2	$55.44	$71.26	$108.01	100.00%	100.00%
m1.large	Linux/Unix	Shared	eu-west-1b	Heavy	15	$1,663.20	$1,726.49	$3,417.87	95.00%	100.00%
m1.small	Linux/Unix	Shared	eu-west-1a	Medium	10	$253.70	$293.25	$366.37	74.00%	100.00%
m1.medium	Linux/Unix	Shared	eu-west-1a	Heavy	4	$151.76	$172.96	$387.08	100.00%	100.00%

This Reserved Instance utilization report displays the average utilization of Reserved Instances in the last two months. This report reveals the following information about the account's Reserved Instances and how they have been utilized.

- Average Utilization

  Most of the Reserved Instances in the table were utilized well. Standouts were the two m1.medium medium utilization Reserved Instances (row 2), which were utilized all the time at 100% average utilization, and the m1.xlarge (row 3) and m1.small (row 4) heavy utilization Reserved Instances, which also were utilized all the time. In contrast, the high-count heavy utilization Reserved Instances (row 5) had lower average utilization rates.

  It is also worth noting that the 12 m1.large medium utilization Reserved Instances (row 1) were utilized on average only 27 percent of the time.
- Maximum Utilization

  At some point during the two-month period, all of the Reserved Instances were used 100 percent.
- Savings

  All across the board, the report shows that for this test account, using Reserved Instances instead of On-Demand instances results in savings for the account owner.
- Question

  Does the account have too many m1.large medium utilization Reserved Instances (row 1)?

Bookmarking a Customized Report

You might want to generate a customized report again. Do this by bookmarking the report.

To bookmark a custom report

1. Select the options and filters for your report. Each selection you make adds a parameter to the console URL. For example, `granularity=Hourly` and `Filters=`*filter_list*.
2. Using your browser, add the console URL as a bookmark.
3. To generate the same report in the future, use the bookmark that you created.

Exporting Your Usage Data

You might want to include your report graph or table in other reports. Do this by exporting the data.

To export usage data

1. Select the options and filters for your report.
2. To export the usage data from the table as a `.csv` file, click **Download** and select **CSV Only**.
3. To export the graphical usage data as a `.png` file, click **Download** and select **Graph Only**.

Options Reference

Use the **Show** options to specify the metric to be displayed by the report graph.

- Average Utilization

  Shows the average of the utilization rates for each hour over the selected time range, where the utilization rate of a bucket for an hour is the number of instance hours used for that hour divided by the total number of Reserved Instances owned in that hour.
- Maximum Utilization

  Shows the highest of the utilization rates of any hour over the selected time range, where the utilization rate of a bucket for an hour is the number of instance hours used for that hour divided by the total number of Reserved Instances owned in that hour.
- Total Cost

  Shows the usage cost plus the amortized portion of the upfront cost of the Reserved Instances in the bucket over the period for which the report is generated.
- Usage Cost

  Shows the total cost based on hourly fees for a selected bucket of Reserved Instances.

Use **Time range** to specify the period on which the report will be based.

> **Note**
> All times are specified in UTC time.

- Last 7 Days

  Shows data for usage that took place during the current and previous six calendar days. Can be used with daily or monthly granularities.
- Last 14 Days

  Shows data for usage that took place during the current and previous 13 calendar days. Can be used with daily or monthly granularities.
- This Month

Shows data for usage that took place during the current calendar month. Can be used with daily or monthly granularities.

- Last 3 Months

  Shows data for usage that took place during the current and previous two calendar months. Can be used with daily or monthly granularities.

- Last 6 Months

  Shows data for usage that took place during the current and previous five calendar months. Can be used with monthly granularities.

- Last 12 Months

  Shows data for usage that took place during the current and previous 11 calendar months. Can be used with monthly granularity.

- This Year

  Shows data for usage that took place during the current calendar year. Can be used with monthly granularity.

- Last 3 Years

  Shows data for usage that took place during the current and previous two calendar years. Can be used with monthly granularity.

- Custom

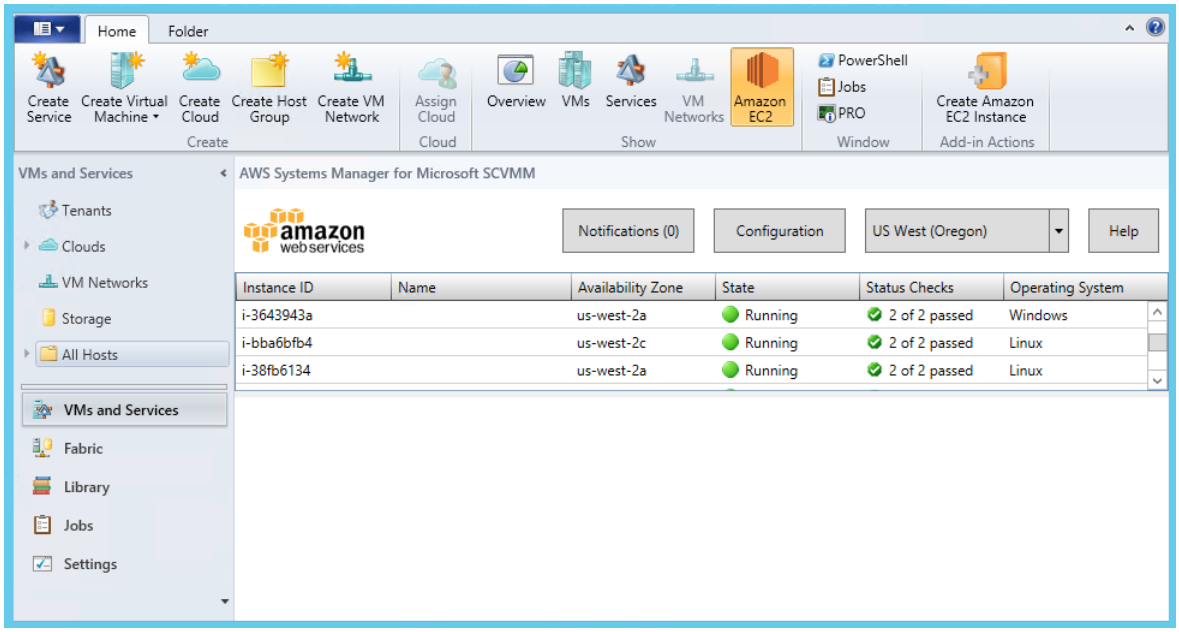
  Shows data for the time range for the entered **Start** and **End** dates specified in the following format: mm/dd/yyyy. Can be used with hourly, daily, or monthly granularities, but you can only specify a maximum time range of two days for hourly data, two months for daily data, and three years for monthly data.

Use **Filter** to scope the data displayed in the report.

- Regions
- Instance Type
- Accounts
- Platforms
- Tenancy
- Offering Types

AWS Systems Manager for Microsoft System Center VMM

Amazon Web Services (AWS) Systems Manager for Microsoft System Center Virtual Machine Manager (SCVMM) provides a simple, easy-to-use interface for managing AWS resources, such as EC2 instances, from Microsoft SCVMM. It is implemented as an add-in for the VMM console. For more information, see AWS Add-ins for Microsoft System Center.



Features

- Administrators can grant permissions to users so that they can manage EC2 instances from SCVMM.
- Users can launch, view, reboot, stop, start, and terminate instances, if they have the required permissions.
- Users can get the passwords for their Windows instances and connect to them using RDP.

- Users can get the public DNS names for their Linux instances and connect to them using SSH.
- Users can import their Hyper-V Windows virtual machines from SCVMM to Amazon EC2.

Limitations

- Users must have an account that they can use to log in to SCVMM.
- You can't launch EC2 instances into EC2-Classic; you must launch them into a VPC.
- You can't import Linux virtual machines from SCVMM to Amazon EC2.
- This is not a comprehensive tool for creating and managing AWS resources. The add-in enables SCVMM users to get started quickly with the basic tasks for managing their EC2 instances. Future releases might support managing additional AWS resources.

Requirements

- An AWS account
- Microsoft System Center VMM 2012 R2 or System Center VMM 2012 SP1 with the latest update roll-up

Getting Started

To get started, see the following documentation:

- Setting Up (p. 631)
- Managing EC2 Instances (p. 635)
- Troubleshooting (p. 642)

Setting Up AWS Systems Manager for Microsoft SCVMM

When you set up AWS Systems Manager, users in your organization can access your AWS resources. The process involves creating accounts, deploying the add-in, and providing your credentials.

Tasks

- Sign Up for AWS (p. 631)
- Set Up Access for Users (p. 632)
- Deploy the Add-In (p. 634)
- Provide Your AWS Credentials (p. 634)

Sign Up for AWS

When you sign up for Amazon Web Services, your AWS account is automatically signed up for all services in AWS. You are charged only for the services that you use.

If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

To sign up for an AWS account

1. Open http://aws.amazon.com/, and then click **Sign Up**.
2. Follow the on-screen instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Set Up Access for Users

The first time that you use AWS Systems Manager, you must provide AWS credentials. To enable multiple users to access the same AWS account using unique credentials and permissions, create an IAM user for each user. You can create one or more groups with policies that grant permissions to perform limited tasks. Then you can create one or more IAM users, and add each user to the appropriate group.

To create an Administrators group

1. Open the IAM console.
2. In the navigation pane, click **Groups** and then click **Create New Group**.
3. In the **Group Name** box, specify **`Administrators`** and then click **Next Step**.
4. On the **Attach Policy** page, select the **AdministratorAccess** AWS managed policy.
5. Click **Next Step** and then click **Create Group**.

To create a group with limited access to Amazon EC2

1. Open the IAM console.
2. In the navigation pane, click **Groups** and then click **Create New Group**.
3. In the **Group Name** box, specify a meaningful name for the group and then click **Next Step**.
4. On the **Attach Policy** page, do not select an AWS managed policy — click **Next Step**, and then click **Create Group**.
5. Click the name of the group you've just created, and then click **Inline Policies**, and then **click here**.
6. Select the **Custom Policy** radio button and then click **Select**.
7. Enter a name for the policy and a policy document that grants limited access to Amazon EC2, and then click **Apply Policy**. For example, you can specify one of the following custom policies.

   Grant users in this group permission to view information about EC2 instances only

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:Describe*",
        "iam:ListInstanceProfiles"
      ],
      "Resource": "*"
    }
  ]
}
```

   Grant users in this group permission to perform all operations on EC2 instances that are supported by the add-in

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:ListInstanceProfiles", "iam:PassRole",
        "ec2:Describe*", "ec2:CreateKeyPair",
        "ec2:CreateTags", "ec2:DeleteTags",
        "ec2:RunInstances", "ec2:GetPasswordData",
        "ec2:RebootInstances", "ec2:StartInstances",
        "ec2:StopInstances", "ec2:TerminateInstances"
      ],
      "Resource": "*"
    }
  ]
}
```

Grant users in this group permission to import a VM to Amazon EC2

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets", "s3:CreateBucket",
        "s3:DeleteBucket", "s3:DeleteObject",
        "s3:GetBucketLocation", "s3:GetObject",
        "s3:ListBucket", "s3:PutObject",
        "ec2:DescribeTags", "ec2:CancelConversionTask",
        "ec2:DescribeConversionTasks", "ec2:DescribeInstanceAttribute",
        "ec2:CreateImage", "ec2:AttachVolume",
        "ec2:ImportInstance", "ec2:ImportVolume",
        "dynamodb:DescribeTable", "dynamodb:CreateTable",
        "dynamodb:Scan", "dynamodb:PutItem", "dynamodb:UpdateItem"
      ],
      "Resource": "*"
    }
  ]
}
```

To create an IAM user, get the user's AWS credentials, and grant the user permissions

1. In the navigation pane, click **Users** and then click **Create New Users**.
2. In box **1**, specify a user name and then click **Create**.
3. Click **Download Credentials** and save the AWS credentials for this IAM user in a secure place. You will need these credentials to access the AWS Systems Manager. After you have downloaded your credentials, click **Close**.
4. Select the user that you just created.
5. Under **Groups** section, click **Add User to Groups**.
6. Select the appropriate group and then click **Add to Groups**.

7. (Optional) If this user must also access the AWS Management Console, you must create a password. Under **Security Credentials**, under **Sign-In Credentials**, click **Manage Password**. Follow the directions to create a password for this IAM user.

Deploy the Add-In

Add-ins for System Center VMM are distributed as `.zip` files. To deploy the add-in, use the following procedure.

To deploy the add-in

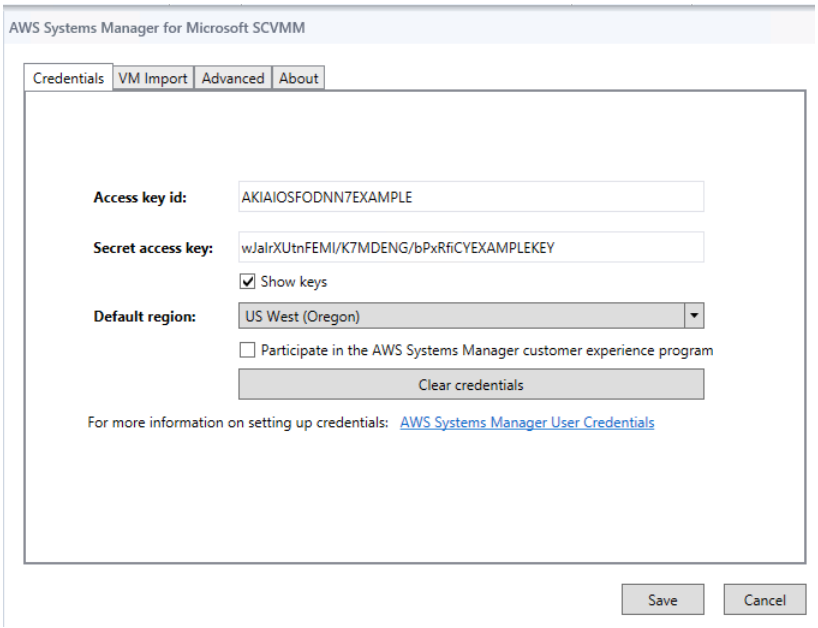
1. From your instance, go to AWS Systems Manager for Microsoft System Center Virtual Machine Manager and click **SCVMM**. Save the `aws-systems-manager-1.5.zip` file to your instance.
2. Open the VMM console.
3. In the navigation pane, click **Settings** and then click **Console Add-Ins**.
4. On the ribbon, click **Import Console Add-in**.
5. On the **Select an Add-in** page, click **Browse** and select the `aws-systems-manager-1.5.zip` file for the add-in that you downloaded.
6. Ignore any warnings that there are assemblies in the add-in that are not signed by a trusted authority. Select **Continue installing this add-in anyway** and then click **Next**.
7. On the **Summary** page, click **Finish**.
8. When the add-in is imported, the status of the job is `Completed`. You can close the **Jobs** window.

Provide Your AWS Credentials

When you use the AWS Systems Manager for the first time, you must provide your AWS credentials. Your access keys identify you to AWS. There are two types of access keys: access key IDs (for example, AKIAIOSFODNN7EXAMPLE) and secret access keys (for example, wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY). You should have stored your access keys in a safe place when you received them.

To provide your AWS credentials

1. Open the VMM console.
2. In the navigation pane, click **VMs and Services**.
3. On the ribbon, click **Amazon EC2**.
4. On the **Credentials** tab, specify your AWS credentials, select a default region, and then click **Save**.



To change these credentials at any time, click **Configuration**.

Managing EC2 Instances Using AWS Systems Manager for Microsoft SCVMM

After you log in to the AWS Systems Manager using your AWS credentials, you can manage your EC2 instances.

Tasks

- Creating an EC2 Instance (p. 635)
- Viewing Your Instances (p. 637)
- Connecting to Your Instance (p. 638)
- Rebooting Your Instance (p. 638)
- Stopping Your Instance (p. 639)
- Starting Your Instance (p. 639)
- Terminating Your Instance (p. 639)

Creating an EC2 Instance

The permissions that you've been granted by your administrator determine whether you can create instances.

Prerequisites

- A virtual private cloud (VPC) with a subnet in the Availability Zone where you'll launch the instance. For more information about creating a VPC, see the Amazon VPC Getting Started Guide.

To create an EC2 instance

1. Open SCVMM.
2. On the ribbon, click **Create Amazon EC2 Instance**.
3. Complete the **Create Amazon EC2 Instance** dialog box as follows:

   a. Select a region for your instance. By default, we select the region that you configured as your default region.

   b. Select a template (known as an AMI) for your instance. To use an AMI provided by Amazon, select **Windows** or **Linux** and then select an AMI from **Image**. To use an AMI that you created, select **My images** and then select the AMI from **Image**.

   c. Select an instance type for the instance. First, select one of the latest instance families from **Family**, and then select an instance type from **Instance type**. To include previous generation instance families in the list, select **Show previous generations.** For more information, see Amazon EC2 Instances and Previous Generation Instances.

   d. Create or select a key pair. To create a key pair, select `Create a new key pair` from **Key pair name** and enter a name for the key pair in the highlighted field (for example, `my-key-pair`).

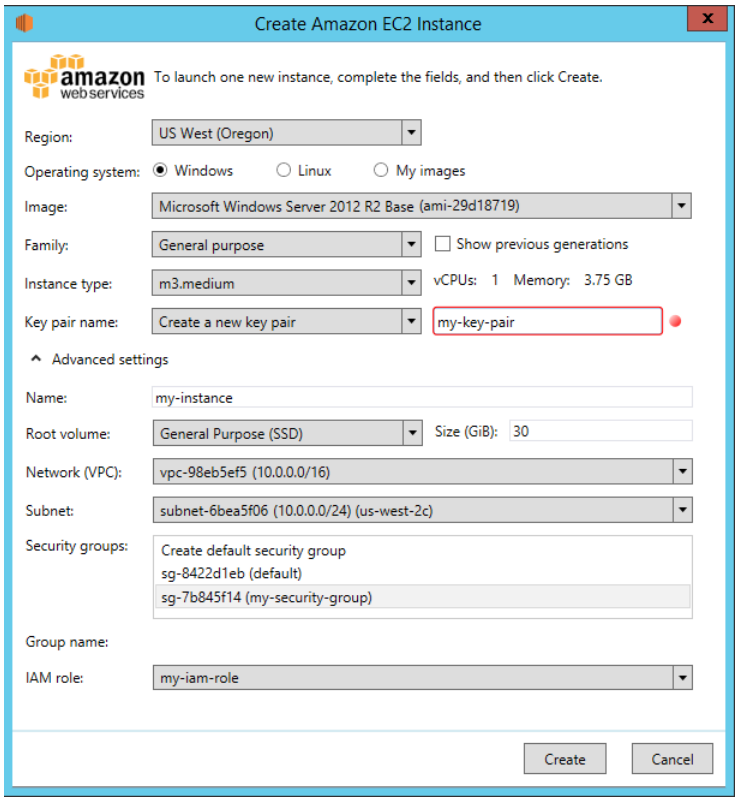
   e. (Optional) Under **Advanced settings**, specify a display name for the instance.

   f. (Optional) Under **Advanced settings**, select a VPC from **Network (VPC)**. Note that this list includes all VPCs for the region, including VPCs created using the Amazon VPC console and the default VPC (if it exists). If you have a default VPC in this region, we select it by default. If the text is "There is no VPC available for launch or import operations in this region", then you must create a VPC in this region using the Amazon VPC console.

   g. (Optional) Under **Advanced settings**, select a subnet from **Subnet**. Note that this list includes all subnets for the selected VPC, including any default subnets. If this list is empty, you must add a subnet to the VPC using the Amazon VPC console, or select a different VPC. Otherwise, we select a subnet for you.

   h. (Optional) Under **Advanced settings**, create a security group or select one or more security groups. If you select `Create default security group`, we create a security group that grants RDP and SSH access to everyone, which you can modify using the Amazon EC2 or Amazon VPC console. You can enter a name for this security group in the **Group name** box.

   i. (Optional) Under **Advanced settings**, select an IAM role. If this list is empty, you can create a role using the IAM console.



4. Click **Create**. If you are creating a key pair, you are prompted to save the `.pem` file. Save this file in a secure place; you'll need it to log in to your instance. You'll receive confirmation that the instance has launched. Click **Close**.

After you've created your instance, it appears in the list of instances for the region in which you launched it. Initially, the status of the instance is `pending`. After the status changes to `running`, your instance is ready for use.

You can manage the lifecycle of your instance using AWS Systems Manager, as described on this page. To perform other tasks, such as the following, you must use the AWS Management Console:

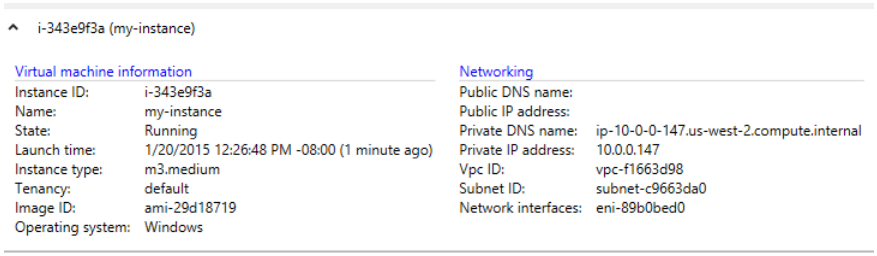
- Attach an Amazon EBS volume to your instance (p. 529)
- Associate an Elastic IP address with your instance (p. 489)
- Enable termination protection (p. 226)

Viewing Your Instances

The permissions that your administrator grants you determine whether you can view instances and get detailed information about them.

To view your instances and get detailed information

1. Open AWS Systems Manager.
2. From the region list, select a region.
3. From the list of instances, select one or more instances.
4. In the lower pane, click the down arrow next to each instance to view detailed information about the instance.



Connecting to Your Instance

You can log in to an EC2 instance if you have the private key (`.pem` file) for the key pair that was specified when launching the instance. The tool that you'll use to connect to your instance depends on whether the instance is a Windows instance or a Linux instance.

To connect to a Windows EC2 instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance, right-click, and then click **Retrieve Windows Password**.
3. In the **Retrieve Default Windows Administrator Password** dialog box, click **Browse**. Select the private key file for the key pair and then click **Open**.
4. Click **Decrypt Password**. Save the password or copy it to the clipboard.
5. Select the instance, right-click, and then click **Connect via RDP**. When prompted for credentials, use the name of the administrator account and the password that you saved in the previous step.
6. Because the certificate is self-signed, you might get a warning that the security certificate is not from a trusted certifying authority. Click **Yes** to continue.

   If the connection fails, see Troubleshooting Windows Instances in the *Amazon EC2 User Guide for Microsoft Windows Instances*.

To connect to a Linux EC2 instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance.
3. In the lower pane, click the down arrow next to the instance ID to view detailed information about the instance.
4. Locate the public DNS name. You'll need this information to connect to your instance.
5. Connect to the instance using PuTTY. For step-by-step instructions, see Connect to Your Linux Instance from Windows Using PuTTY in the *Amazon EC2 User Guide for Linux Instances*.

Rebooting Your Instance

The permissions that you've been granted by your administrator determine whether you can reboot instances.

To reboot your instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance.
3. Right-click the instance, and then click **Reset (Reboot)**.

4. When prompted for confirmation, click **Yes**.

Stopping Your Instance

The permissions that you've been granted by your administrator determine whether you can stop instances.

To stop your instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance.
3. Right-click the instance, and then click **Shut Down (Stop)**.
4. When prompted for confirmation, click **Yes**.

Starting Your Instance

The permissions that you've been granted by your administrator determine whether you can start instances.

To start your instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance.
3. Right-click the instance, and then click **Power On (Start)**.
4. When prompted for confirmation, click **Yes**.

If you get a quota error when you try to start an instance, you have reached your concurrent running instance limit. The default limit for your AWS account is 20. If you need additional running instances, complete the form at Request to Increase Amazon EC2 Instance Limit.

Terminating Your Instance

The permissions that you've been granted by your administrator determine whether you can terminate instances.

To terminate your instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance.
3. Right-click the instance, and then click **Delete (Terminate)**.
4. When prompted for confirmation, click **Yes**.

Importing Your Virtual Machine Using AWS Systems Manager for Microsoft SCVMM

You can launch an EC2 instance from a virtual machine that you import from SCVMM to Amazon EC2.

> **Important**
> You can't import Linux virtual machines from SCVMM to Amazon EC2.

Contents

- Prerequisites (p. 640)
- Importing Your Virtual Machine (p. 640)
- Checking the Import Task Status (p. 641)
- Backing Up Your Imported Instance (p. 641)

Prerequisites

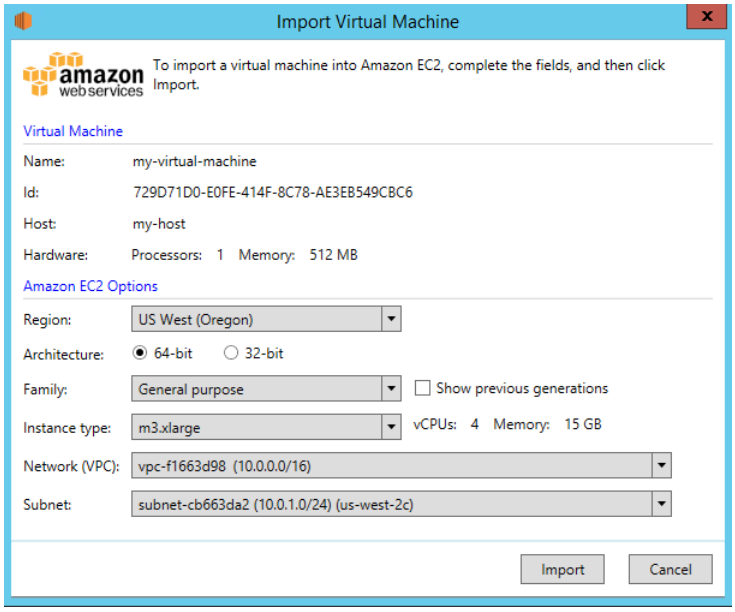
- Ensure that your VM is ready. For more information, see Prepare Your VM (p. 187).
- In AWS Systems Manager, click **Configuration**, select the **VM Import** tab, and review the following settings:
  - **S3 bucket prefix**: We create a bucket for disk images to be uploaded before they are imported. The name of the bucket starts with the prefix listed here and includes the region (for example, `us-west-2`). To delete the disk images after they are imported, select **Clean up S3 bucket after import**.
  - **VM image export path**: A location for the disk images exported from the VM. To delete the disk images after they are imported, select **Clean up export path after import**.
  - **Alternate Hyper-V PowerShell module path**: The location of the Hyper-V PowerShell module, if it's not installed in the standard location. For more information, see Installing the Hyper-V Management Tools in the Microsoft TechNet Library.

Importing Your Virtual Machine

The permissions that you've been granted by your administrator determine whether you can import HyperV Windows virtual machines from SCVMM to AWS.

To import your virtual machine

1. Open SCVMM.
2. On the ribbon, click **VMs**. Select your virtual machine from the list.
3. On the ribbon, click **Import VM to Amazon EC2**.
4. Complete the **Import Virtual Machine** dialog box as follows:
   a. Select a region for the instance. By default, we select the region that you configured as your default region.
   b. Select an instance type for the instance. First, select one of the latest instance families from **Family**, and then select an instance type from **Instance type**. To include previous generation instance families in the list, select **Show previous generations.** For more information, see Amazon EC2 Instances and Previous Generation Instances.
   c. Select a VPC from **Network (VPC)**. Note that this list includes all VPCs for the region, including VPCs created using the Amazon VPC console and the default VPC (if it exists). If you have a default VPC in this region, we select it by default. If the text is "There is no VPC available for launch or import operations in this region", then you must create a VPC in this region using the Amazon VPC console.
   d. Select a subnet from **Subnet**. Note that this list includes all subnets for the selected VPC, including any default subnets. If this list is empty, you must add a subnet to the VPC using the Amazon VPC console, or select a different VPC. Otherwise, we select a subnet for you.



5. Click **Import**. If you haven't specified the required information in the **VM Import** tab, you'll receive an error asking you to provide the required information. Otherwise, you'll receive confirmation that the import task has started. Click **Close**.

Checking the Import Task Status

The import task can take several hours to complete. To view the current status, open AWS System Manager and click **Notifications**.

You'll receive the following notifications as the import task progresses:

- `Import VM: Created Import VM Task`
- `Import VM: Export VM Disk Image Done`
- `Import VM: Upload to S3`
- `Import VM: Image Conversion Starting`
- `Import VM: Image Conversion Done`
- `Import VM: Import Complete`

Note that you'll receive the `Import VM: Upload to S3`, `Import VM: Image Conversion Starting`, and `Import VM: Image Conversion Done` notifications for each disk image converted.

If the import task fails, you'll receive the notification `Import VM: Import Failed`. For more information about troubleshooting issues with import tasks, see Errors Importing a VM (p. 643).

Backing Up Your Imported Instance

After the import operation completes, the instance runs until it is terminated. If your instance is terminated, you can't connect to or recover the instance. To ensure that you can start a new instance with the same software as an imported instance if needed, create an Amazon Machine Image (AMI) from the imported instance. For more information, see Creating an Amazon EBS-Backed Windows AMI (p. 68).

Troubleshooting AWS Systems Manager for Microsoft SCVMM

The following are common errors and troubleshooting steps.

Contents

- Error: Add-in cannot be installed (p. 642)
- Installation Errors (p. 642)
- Checking the Log File (p. 643)
- Errors Importing a VM (p. 643)
- Uninstalling the Add-In (p. 643)

Error: Add-in cannot be installed

If you receive the following error, try installing KB2918659 on the computer running the VMM console. For more information, see Description of System Center 2012 SP1 Update Rollup 5. Note that you don't need to install all the updates listed in this article to address this issue, just KB2918659.

```
Add-in cannot be installed
The assembly "Amazon.Scvmm.Addin" referenced to by add-in component "AWS Systems
 Manager for
Microsoft SCVMM" could not be found in the add-in package. This could be due
to the following
reasons:
1. The assembly was not included with the add-in package.
2. The AssemblyName attribute for the add-in does not match the name of the
add-in assembly.
3. The assembly file is corrupt and cannot be loaded.
```

Installation Errors

If you receive one of the following errors during installation, it is likely due to an issue with SCVMM:

```
Could not update managed code add-in pipeline due to the following error:
Access to the path 'C:\Program Files\Microsoft System Center 2012\Virtual Machine
 Manager
\Bin\AddInPipeline\PipelineSegments.store' is denied.
```

```
Could not update managed code add-in pipeline due to the following error:
The required folder 'C:\Program Files\Microsoft System Center 2012\Virtual Ma
chine Manager
\Bin\AddInPipeline\HostSideAdapters' does not exist.
```

```
Add-in cannot be installed
The assembly "Microsoft.SystemCenter.VirtualMachineManager.UIAddIns.dll" refer
enced by the
add-in assembly "Amazon.Scvmm.AddIn" could not be found in the add-in package.
 Make sure
that this assembly was included with the add-in package.
```

Try one of the following steps to work around this issue:

- Grant authenticated users permission to read and execute the `C:\Program Files\Microsoft System Center 2012\Virtual Machine Manager\Bin\AddInPipeline` folder. In Windows Explorer, right-click the folder, select **Properties**, and then select the **Security** tab.
- Close the SCVMM console and start it one time as an administrator. From the **Start** menu, locate SCVMM, right-click, and then select **Run as administrator**.

Checking the Log File

If you have a problem using the add-in, check the generated log file, `%APPDATA%\Amazon\SCVMM\ec2addin.log`, for useful information.

Errors Importing a VM

The log file, `%APPDATA%\Amazon\SCVMM\ec2addin.log`, contains detailed information about the status of an import task. The following are common errors that you might see in the log file when you import your VM from SCVMM to Amazon EC2.

Error: Unable to extract Hyper-V VirtualMachine object

Solution: Configure the path to the Hyper-V PowerShell module.

Error: You do not have permission to perform the operation

Solution: Contact your administrator.

Uninstalling the Add-In

If you need to uninstall the add-in, use the following procedure.

To uninstall the add-in

1. Open the VMM console.
2. Select the **Settings** workspace, and then click **Console Add-Ins**.
3. Select **AWS Systems Manager for Microsoft SCVMM**.
4. On the ribbon, click **Remove**.
5. When prompted for confirmation, click **Yes**.

If you reinstall the add-in after uninstalling it and receive the following error, delete the path as suggested by the error message.

```
Error (27301)
There was an error while installing the add-in. Please ensure that the following
 path does not
exist and then try the installation again.

C:\Program Files\Microsoft System Center 2012\Virtual Machine Manager\Bin\AddIn
Pipeline\
AddIns\EC2WINDOWS...
```

AWS Management Pack for Microsoft System Center

Amazon Web Services (AWS) offers a complete set of infrastructure and application services for running almost anything in the cloud—from enterprise applications and big data projects to social games and mobile apps. The AWS Management Pack for Microsoft System Center provides availability and performance monitoring capabilities for your applications running in AWS.

The AWS Management Pack allows Microsoft System Center Operations Manager to access your AWS resources (such as instances and volumes), so that it can collect performance data and monitor your AWS resources. The AWS Management Pack is an extension to System Center Operations Manager. There are two versions of the AWS Management Pack: one for System Center 2012 — Operations Manager and another for System Center Operations Manager 2007 R2.

The AWS Management Pack uses Amazon CloudWatch metrics and alarms to monitor your AWS resources. Amazon CloudWatch metrics appear in Microsoft System Center as performance counters and Amazon CloudWatch alarms appear as alerts.

You can monitor the following resources:

- EC2 instances
- EBS volumes
- ELB load balancers
- Auto Scaling groups and Availability Zones
- Elastic Beanstalk applications
- CloudFormation stacks
- CloudWatch Alarms
- CloudWatch Custom Metrics

Contents

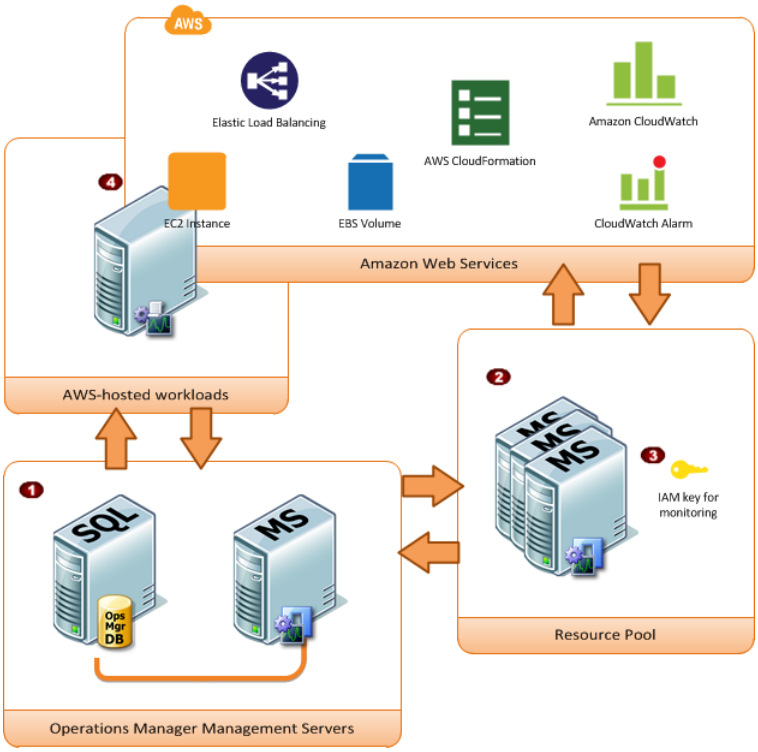
- Overview of AWS Management Pack for System Center 2012 (p. 645)
- Overview of AWS Management Pack for System Center 2007 R2 (p. 646)
- Downloading the AWS Management Pack (p. 647)
- Deploying the AWS Management Pack (p. 648)

- Using the AWS Management Pack (p. 658)
- Upgrading the AWS Management Pack (p. 677)
- Uninstalling the AWS Management Pack (p. 678)
- Troubleshooting the AWS Management Pack (p. 678)

Overview of AWS Management Pack for System Center 2012

The AWS Management Pack for System Center 2012 — Operations Manager uses a resource pool that contains one or more management servers to discover and monitor your AWS resources. You can add management servers to the pool as you increase the number of AWS resources that you use.

The following diagram shows the main components of AWS Management Pack.



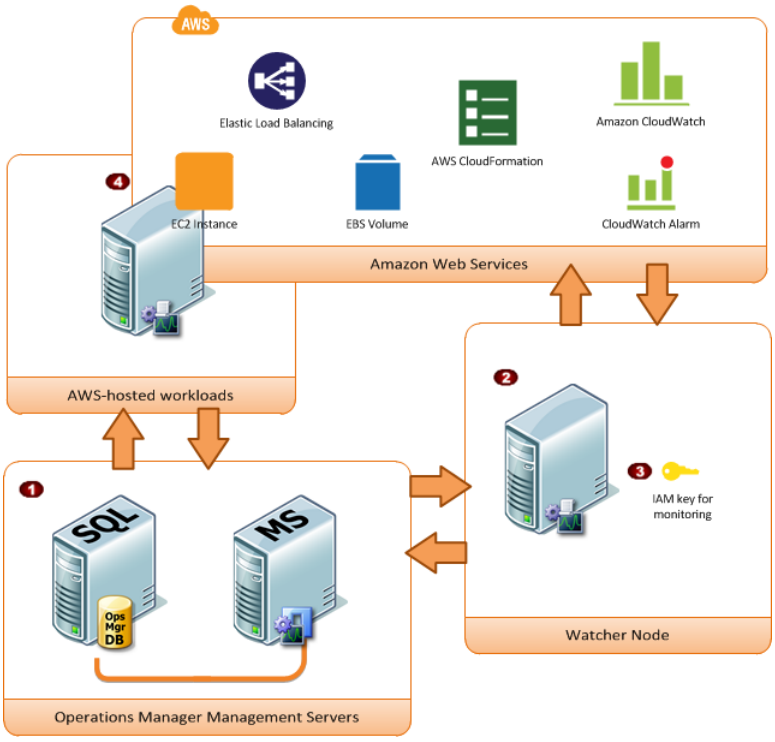
Item	Component	Description
1	Operations Manager infra-structure	One or more management servers and their dependencies, such as Microsoft SQL Server and a Microsoft Active Directory domain. These servers can either be deployed on-premises or in the AWS cloud; both scenarios are supported.
2	Resource pool	One or more management servers used for communicating with AWS using the AWS SDK for .NET. These servers must have Internet connectivity.

Item	Component	Description
3	AWS credentials	An access key ID and a secret access key used by the management servers to make AWS API calls. You must specify these credentials while you configure the AWS Management Pack. We recommend that you create an IAM user with read-only privileges and use those credentials. For more information about creating an IAM user, see Adding a New User to Your AWS Account in the *IAM User Guide*.
4	EC2 instances	Virtual computers running in the AWS cloud. Some instances might have the Operations Manager Agent installed, others might not. When you install Operations Manager Agent you can see the operating system and application health apart from the instance health.

Overview of AWS Management Pack for System Center 2007 R2

The AWS Management Pack for System Center Operations Manager 2007 R2 uses a designated computer that connects to your System Center environment and has Internet access, called a *watcher node*, to call AWS APIs to remotely discover and collect information about your AWS resources.

The following diagram shows the main components of AWS Management Pack.



Item	Component	Description
1	Operations Manager infrastructure	One or more management servers and their dependencies, such as Microsoft SQL Server and a Microsoft Active Directory domain. These servers can either be deployed on-premises or in the AWS cloud; both scenarios are supported.
2	Watcher node	A designated agent-managed computer used for communicating with AWS using the AWS SDK for .NET. It can either be deployed on-premises or in the AWS cloud, but it must be an agent-managed computer, and it must have Internet connectivity. You can use exactly one watcher node to monitor an AWS account. However, one watcher node can monitor multiple AWS accounts. For more information about setting up a watcher node, see Deploying Windows Agents in the Microsoft System Center documentation.
3	AWS credentials	An access key ID and a secret access key used by the watcher node to make AWS API calls. You must specify these credentials while you configure the AWS Management Pack. We recommend that you create an IAM user with read-only privileges and use those credentials. For more information about creating an IAM user, see Adding a New User to Your AWS Account in the *IAM User Guide*.
4	EC2 instances	Virtual computers running in the AWS cloud. Some instances might have the Operations Manager Agent installed, others might not. When you install the Operations Manager Agent you can see the operating system and application health apart from the instance health.

Downloading the AWS Management Pack

To get started, download the AWS Management Pack. The AWS Management Pack is free. You might incur charges for Amazon CloudWatch, depending on how you configure monitoring or how many AWS resources you monitor.

System Center 2012

Before you download the AWS Management Pack, ensure that your systems meet the following system requirements and prerequisites.

System Requirements

- System Center Operations Manager 2012 R2 or System Center Operations Manager 2012 SP1
- Cumulative Update 1 or later. You must deploy the update to the management servers monitoring AWS resources, as well as agents running the watcher nodes and agents to be monitored by the AWS Management Pack. We recommend that you deploy the latest available Operations Manager updates on all computers monitoring AWS resources.
- Microsoft.Unix.Library MP version 7.3.2026.0 or later

Prerequisites

- Your data center must have at least one management server configured with Internet connectivity. The management servers must have the Microsoft .NET Framework version 4.5 or later and PowerShell 2.0 or later installed.
- The action account for the management server must have local administrator privileges on the management server.

To download the AWS Management Pack

1. On the AWS Add-Ins for Microsoft System Center website, click **SCOM 2012**.
2. Save `AWS-SCOM-MP-2.5.zip` to your computer and unzip it.

Continue with Deploying the AWS Management Pack (p. 648).

System Center 2007 R2

Before you download the AWS Management Pack, ensure that your systems meet the following system requirements and prerequisites.

System Requirements

- System Center Operations Manager 2007 R2
- Microsoft.Unix.Library MP version 6.1.7000.256 or later

Prerequisites

- Your data center must have an agent-managed computer with Internet connectivity that you designate as the watcher node. The watcher node must have the following Agent Proxy option enabled: **Allow this agent to act as a proxy and discover managed objects on other computers**. The watcher node must have the Microsoft .NET Framework version 3.5.1 or later and PowerShell 2.0 or later installed.
- The action account for the watcher node must have local administrator privileges on the watcher node.
- You must ensure that your watcher node has the agent installed, has Internet access, and can communicate with the management servers in your data center. For more information, see Deploying Windows Agents in the Microsoft System Center documentation.

To download the AWS Management Pack

1. On the AWS Add-Ins for Microsoft System Center website, click **SCOM 2007**.
2. Save `AWS-MP-Setup-2.5.msi` to your computer.

Continue with Deploying the AWS Management Pack (p. 648).

Deploying the AWS Management Pack

Before you can deploy the AWS Management Pack, you must download it. For more information, see Downloading the AWS Management Pack (p. 647).

Tasks

- Step 1: Installing the AWS Management Pack (p. 649)
- Step 2: Configuring the Watcher Node (p. 650)
- Step 3: Create an AWS Run As Account (p. 651)
- Step 4: Run the Add Monitoring Wizard (p. 653)
- Step 5: Configure Ports and Endpoints (p. 657)

Step 1: Installing the AWS Management Pack

After you download the AWS Management Pack, you must configure it to monitor one or more AWS accounts.

System Center 2012

To install the AWS Management Pack

1. In the Operations console, on the **Go** menu, click **Administration**, and then click **Management Packs**.
2. In the **Actions** pane, click **Import Management Packs**.
3. On the **Select Management Packs** page, click **Add**, and then click **Add from disk**.
4. In the **Select Management Packs to import** dialog box, select the `Amazon.AmazonWebServices.mpb` file from the location where you downloaded it, and then click **Open**.
5. On the **Select Management Packs** page, under **Import list**, select the **Amazon Web Services** management pack, and then click **Install**.

   Note
   System Center Operations Manager doesn't import any management packs in the **Import** list that display an **Error** icon.
6. The **Import Management Packs** page shows the progress for the import process. If a problem occurs, select the management pack in the list to view the status details. Click **Close**.

System Center 2007 R2

To install the AWS Management Pack

The management pack is distributed as a Microsoft System Installer file, `AWS-MP-Setup.msi`. It contains the required DLLs for the watcher node, root management server, and Operations console, as well as the `Amazon.AmazonWebServices.mp` file.

1. Run `AWS-MP-Setup.msi`.

   Note
   If your root management server, Operations console, and watcher node are on different computers, you must run the installer on each computer.
2. On the **Welcome to the Amazon Web Services Management Pack Setup Wizard** screen, click **Next**.
3. On the **End-User License Agreement** screen, read the license agreement, and, if you accept the terms, select the **I accept the terms in the License Agreement** check box, and then click **Next**.
4. On the **Custom Setup** screen, select the features you want to install, and then click **Next**.

   Operations Console
   Installs `Amazon.AmazonWebServices.UI.Pages.dll` and registers it in the Global Assembly Cache (GAC), and then installs `Amazon.AmazonWebServices.mp`.

Root Management Server
Installs `Amazon.AmazonWebServices.Modules.dll`, `Amazon.AmazonWebServices.SCOM.SDK.dll` and the AWS SDK for .NET (`AWSSDK.dll`), and then registers them in the GAC.

AWS Watcher Node
Installs `Amazon.AmazonWebServices.Modules.dll` and `Amazon.AmazonWebServices.SCOM.SDK.dll`, and then installs the AWS SDK for .NET (`AWSSDK.dll`) and registers it in the GAC.

5. On the **Ready to install Amazon Web Services Management Pack** screen, click **Install**.
6. On the **Completed the Amazon Web Services Management Pack Setup Wizard** screen, click **Finish**.

   Note
   The required DLLs are copied and registered in the GAC, and the management pack file (`*.mp`) is copied to the `Program Files (x86)/Amazon Web Services Management Pack` folder on the computer running the Operations console. Next, you must import the management pack into System Center.
7. In the Operations console, on the **Go** menu, click **Administration**, and then click **Management Packs**.
8. In the **Actions** pane, click **Import Management Packs**.
9. On the **Select Management Packs** page, click **Add**, and then click **Add from disk**.
10. In the **Select Management Packs to import** dialog box, change the directory to `C:\Program Files (x86)\Amazon Web Services Management Pack`, select the `Amazon.AmazonWebServices.mp` file, and then click **Open**.
11. On the **Select Management Packs** page, under **Import list**, select the **Amazon Web Services** management pack, and then click **Install**.

    Note
    System Center Operations Manager doesn't import any management packs in the **Import** list that display an **Error** icon.
12. The **Import Management Packs** page shows the progress for the import process. If a problem occurs, select the management pack in the list to view the status details. Click **Close**.

Step 2: Configuring the Watcher Node

On System Center Operations Manager 2007 R2, the watcher node runs discoveries that go beyond the watcher node computer, so you must enable the proxy agent option on the watcher node. The proxy agent allows those discoveries to access the objects on other computers.

Note
If your system is configured with a large number of resources, we recommend that you configure one management server as a Watcher Node. Having a separate Watcher Node management server can improve performance.

If you're using System Center 2012 — Operations Manager, you can skip this step.

To enable the proxy agent on System Center Operations Manager 2007 R2

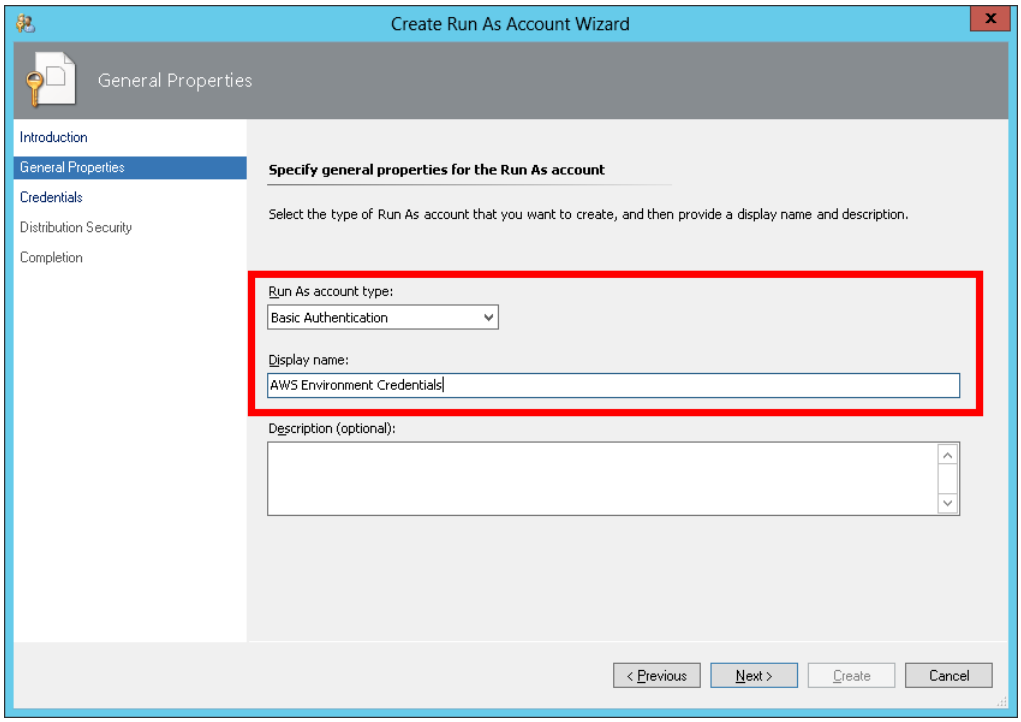
1. In the Operations console, on the **Go** menu, click **Administration**.
2. In the **Administration** workspace, under **Device Management**, click **Agent Managed**.
3. In the **Agent Managed** list, right-click the watcher node, and then click **Properties**.
4. In the **Agent Properties** dialog box, click the **Security** tab, select **Allow this agent to act as proxy and discover managed objects on other computers**, and then click **OK**.

Step 3: Create an AWS Run As Account

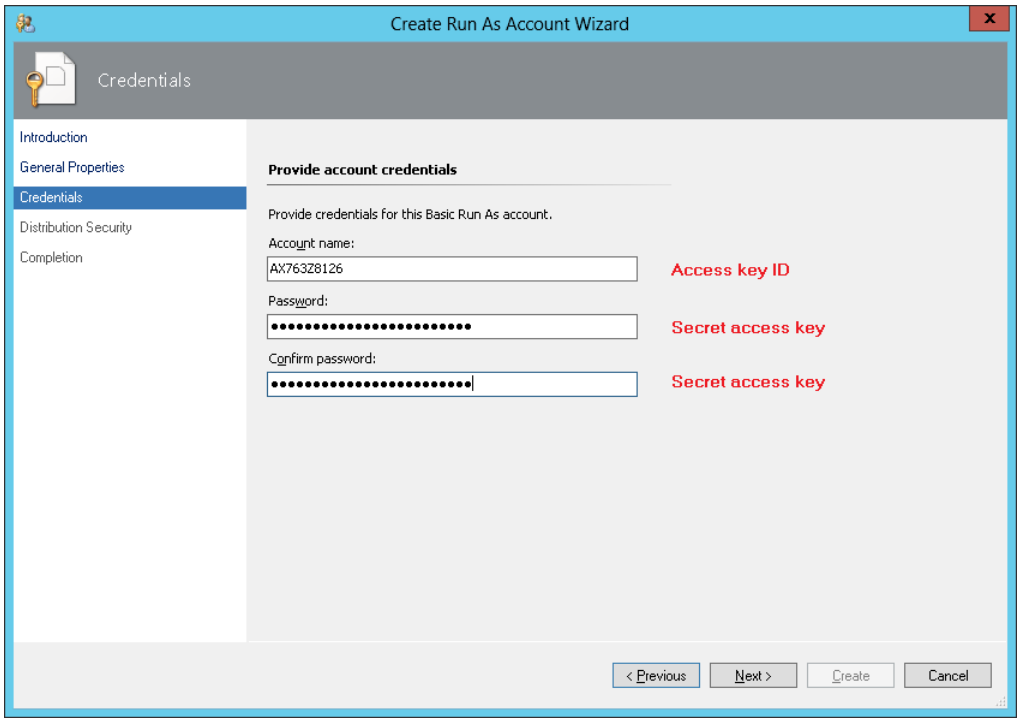
You must set up credentials that grant AWS Management Pack access to your AWS resources.

To create an AWS Run As account

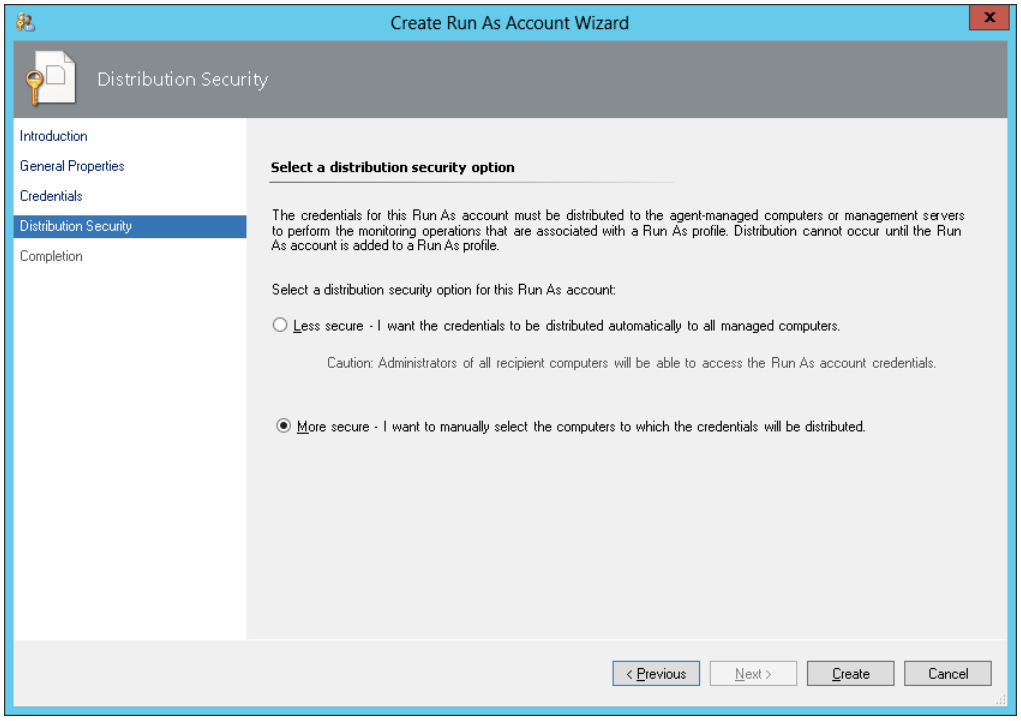
1. We recommend that you create an IAM user with the minimum access rights required (for example, the **ReadOnlyAccess** AWS managed policy works in most cases). You'll need the access keys (access key ID and secret access key) for this user to complete this procedure. For more information, see Administering Access Keys for IAM Users in the *IAM User Guide*.
2. In the Operations console, on the **Go** menu, click **Administration**.
3. In the **Administration** workspace, expand the **Run As Configuration** node, and then select **Accounts**.
4. Right-click the **Accounts** pane, and then click **Create Run As Account**.
5. In the **Create Run As Account Wizard**, on the **General Properties** page, in the **Run As account type** list, select **Basic Authentication**.
6. Enter a display name (for example, "My IAM Account") and a description, and then click **Next**.



7. On the **Credentials** page, enter the access key ID in the **Account name** box and the secret access key in the **Password** box, and then click **Next**.

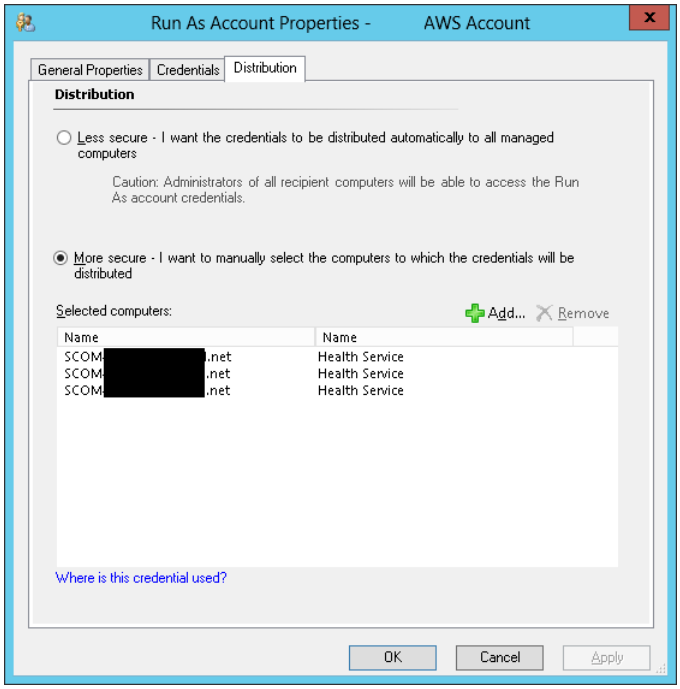


8. On the **Distribution Security** page, select **More secure - I want to manually select the computers to which the credentials will be distributed**, and then click **Create**.



9. Click **Close**.
10. In the list of accounts, select the account that you just created.
11. In the **Actions** pane, click **Properties**.

12. In the **Properties** dialog box, verify that the **More Secure** option is selected and that all management servers to be used to monitor your AWS resources are listed.



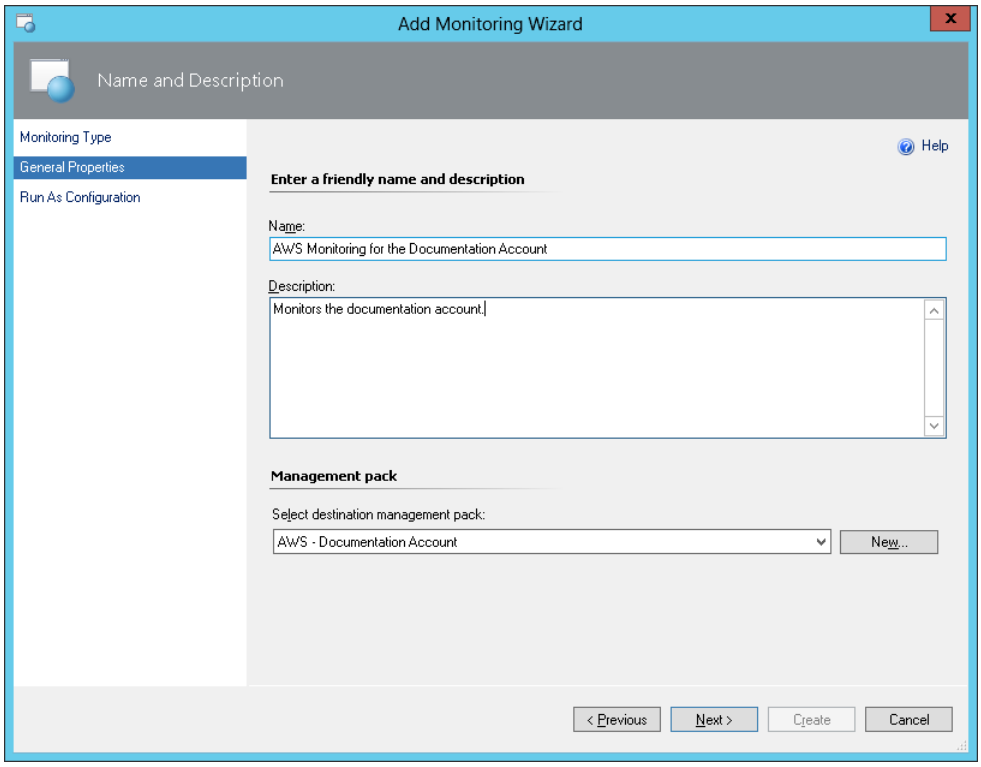
Step 4: Run the Add Monitoring Wizard

You can configure the AWS Management Pack to monitor a particular AWS account by using the Add Monitoring Wizard, which is available in the **Authoring** workspace of the Operations console. This wizard creates a management pack that contains the settings for the AWS account to monitor. You must run this wizard to monitor each AWS account. For example, if you want to monitor two AWS accounts, you must run the wizard twice.

System Center 2012

To run the Add Monitoring Wizard on System Center 2012 — Operations Manager

1. In the Operations console, on the **Go** menu, click **Authoring**.
2. In the **Authoring** workspace, expand the **Management Pack Templates** node, right-click **Amazon Web Services**, and then click **Add Monitoring Wizard**.
3. In the **Add Monitoring Wizard**, in the **Select the monitoring type** list, select **Amazon Web Services**, and then click **Next**.
4. On the **General Properties** page, in the **Name** box, enter a name (for example, "My AWS Resources"). In the **Description** box, enter a description.
5. In the **Select destination management pack** list, select an existing management pack (or click **New** to create one) where you want to save the settings. Click **Next**.



Note

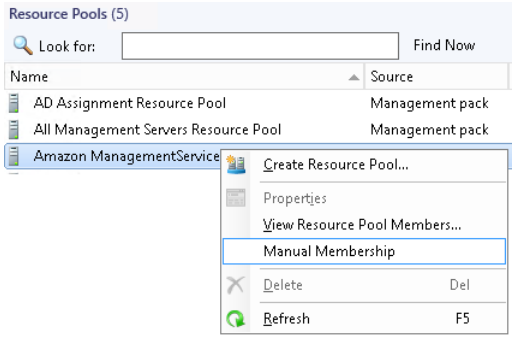
By default, when you create a management pack object, disable a rule or monitor, or create an override, Operations Manager saves the setting to the default management pack. As a best practice, you should create a separate management pack for each sealed management pack that you want to customize, instead of saving your customized settings to the default management pack.

6. The AWS Management Pack automatically creates a resource pool and adds the management servers to it. To control server membership, make the following changes:

   a. Click **Administration** on the **Go** menu.

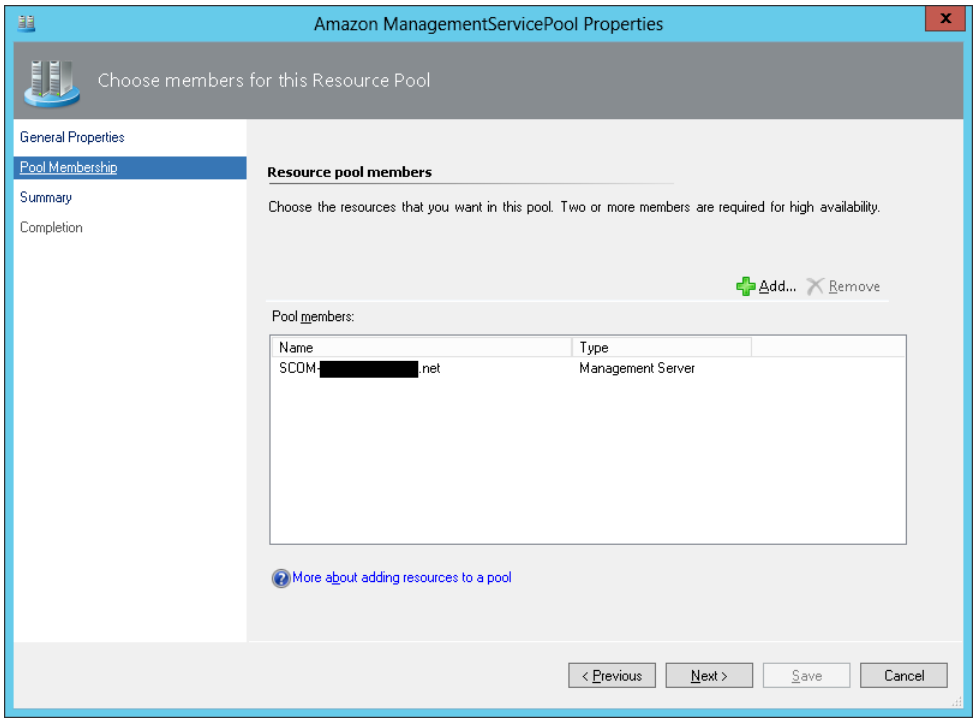
   b. Click the **Resource Pools** node.

   c. Right-click the **AWS Resource Pool** in the **Resource Pools** pane and select **Manual Membership**.

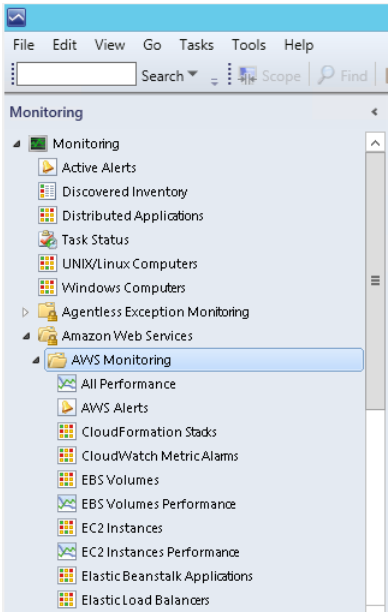


   d. Right-click the **AWS Resource Pool** in the **Resource Pools** pane and select **Properties**.

   e. On the **Pool Membership** page, remove the management servers that should not monitor AWS resources.



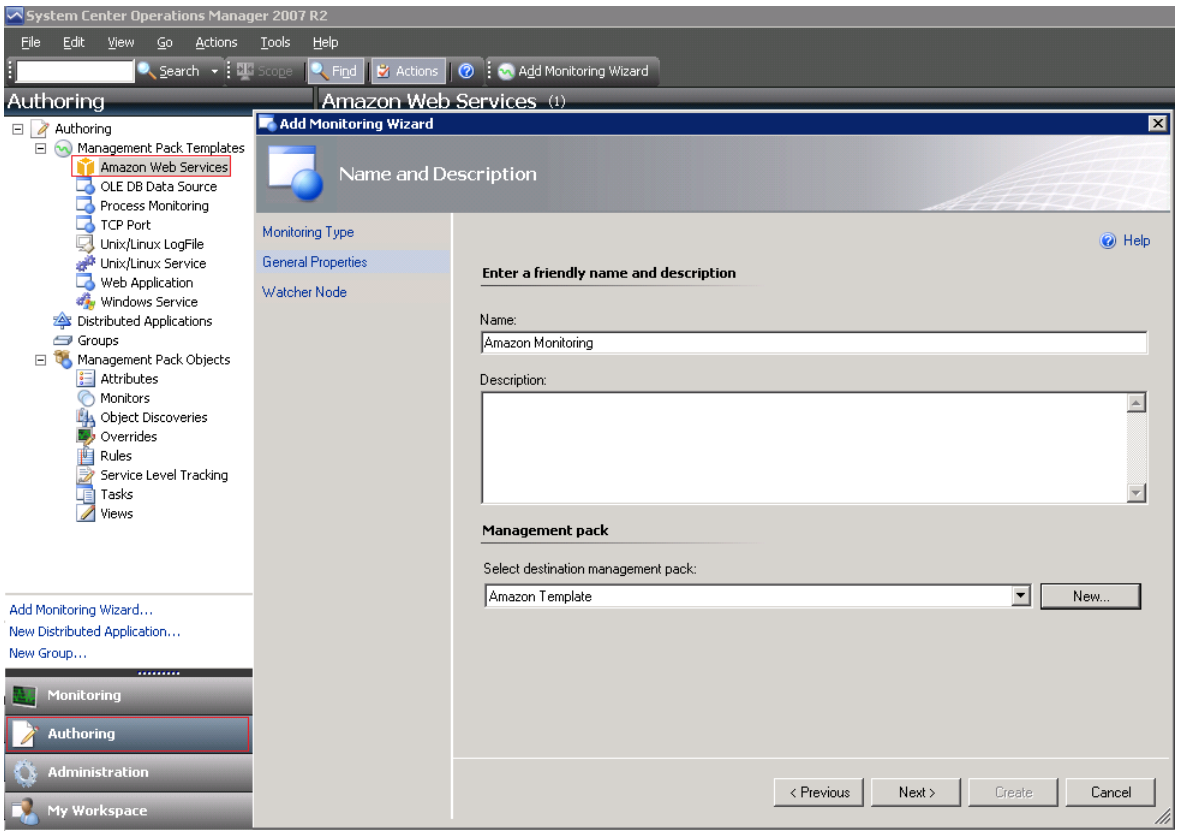
7. After the AWS Management Pack is configured, it shows up as a sub-folder of the `Amazon Web Services` folder in the **Monitoring** workspace of the Operations console.



System Center 2007 R2

To run the Add Monitoring Wizard on System Center Operations Manager 2007

1. In the Operations console, on the **Go** menu, click **Authoring**.
2. In the **Authoring** workspace, expand the **Management Pack Templates** node, right-click **Amazon Web Services**, and then click **Add Monitoring Wizard**.
3. In the **Add Monitoring Wizard**, in the **Select the monitoring type list**, select **Amazon Web Services**, and then click **Next**.
4. On the **General Properties** page, in the **Name** box, enter a name (for example, "My AWS Resources"). In the **Description** box, enter a description.
5. In the **Select destination management pack** drop-down list, select an existing management pack (or click **New** to create a new one) where you want to save the settings. Click **Next**.

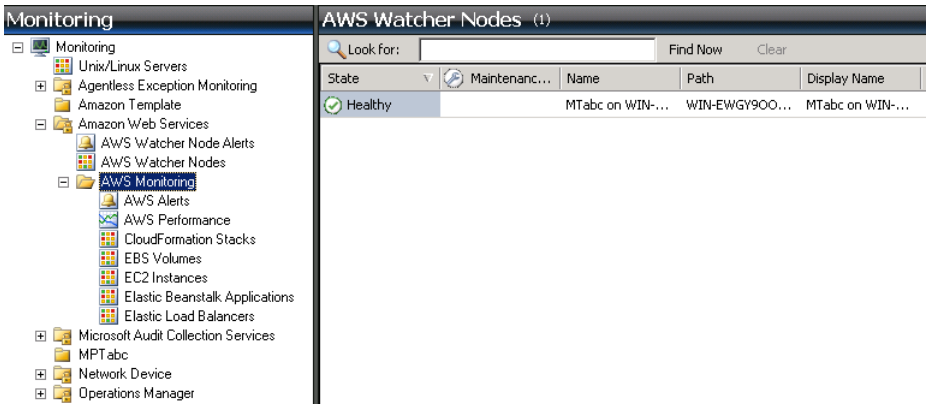


Note
By default, when you create a management pack object, disable a rule or monitor, or create an override, Operations Manager saves the setting to the default management pack. As a best practice, you should create a separate management pack for each sealed management pack that you want to customize, instead of saving your customized settings to the default management pack.

6. On the **Watcher Node Configuration** page, in the **Watcher Node** list, select an agent-managed computer to act as the watcher node.
7. In the **Select AWS Run As account** drop-down list, select the Run As account that you created earlier, and then click **Create**.
8. After the AWS Management Pack is configured, it first discovers the watcher node. To verify that the watcher node was discovered successfully, navigate to the **Monitoring** workspace in the Operations

console. You should see a new `Amazon Web Services` folder and an `Amazon Watcher Nodes` subfolder under it. This subfolder displays the watcher nodes. The AWS Management Pack automatically checks and monitors the watcher node connectivity to AWS. When the watcher node is discovered, it shows up in this list. When the watcher node is ready, its state changes to `Healthy`.

Note
To establish connectivity with AWS, the AWS Management Pack requires that you deploy the AWS SDK for .NET, modules, and scripts to the watcher node. This can take about ten minutes. If the watcher node doesn't appear, or if you see the state as `Not Monitored`, verify your Internet connectivity and IAM permissions. For more information, see Troubleshooting the AWS Management Pack (p. 678).



9. After the watcher node is discovered, dependent discoveries are triggered, and the AWS resources are added to the **Monitoring** workspace of the Operations console.

   Note
   The discovery of AWS resources should finish within twenty minutes. This process can take more time, based on your Operations Manager environment, your AWS environment, the load on the management server, and the load on the watcher node. For more information, see Troubleshooting the AWS Management Pack (p. 678).

Step 5: Configure Ports and Endpoints

The AWS Management Pack for Microsoft System Center must be able to communicate with AWS services to monitor the performance of those services and provide alerts in System Center. For monitoring to succeed, you must configure the firewall on the Management Pack servers to allow outbound HTTP calls on ports 80 and 443 to the AWS endpoints for the following services.

This enables monitoring for the following AWS services:

- Amazon Elastic Compute Cloud (EC2)
- Elastic Load Balancing
- Auto Scaling
- AWS Elastic Beanstalk
- Amazon CloudWatch
- AWS CloudFormation

The AWS Management Pack uses the public APIs in the AWS SDK for .NET to retrieve information from these services over ports 80 and 443. Log on to each server and enable outbound firewall rules for ports 80 and 443.

If your firewall application supports more detailed settings you can configure specific endpoints for each service. An endpoint is a URL that is the entry point for a web service. For example, ec2.us-west-2.amazonaws.com is an entry point for the Amazon EC2 service. To configure endpoints on your firewall, locate the specific endpoint URLs for the AWS services you are running and specify those endpoints in your firewall application.

Using the AWS Management Pack

You can use the AWS Management Pack to monitor the health of your AWS resources.

Contents

- Views (p. 658)
- Discoveries (p. 672)
- Monitors (p. 673)
- Rules (p. 673)
- Events (p. 674)
- Health Model (p. 675)
- Customizing the AWS Management Pack (p. 676)

Views

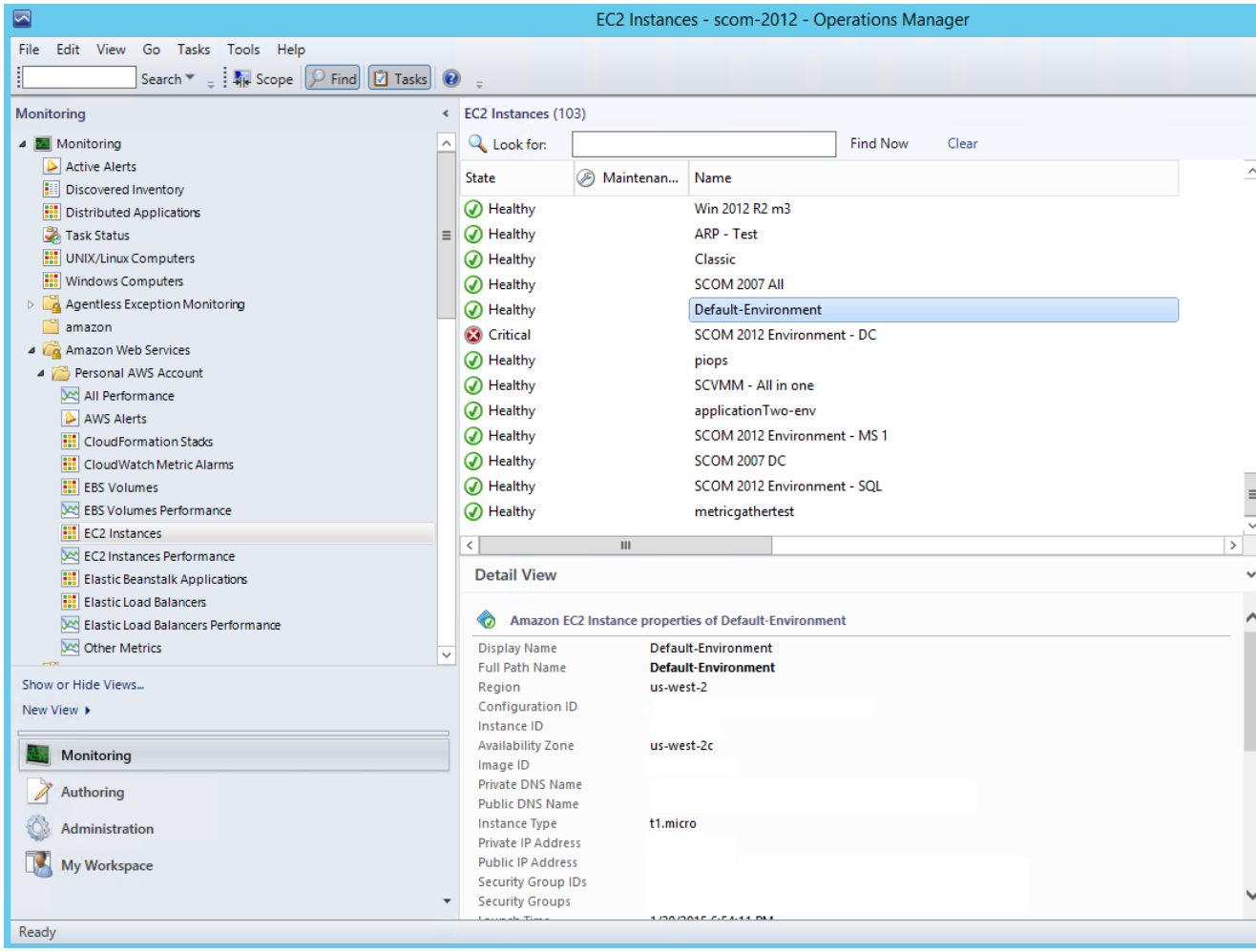
The AWS Management Pack provides the following views, which are displayed in the **Monitoring** workspace of the Operations console.

Views

- EC2 Instances (p. 658)
- Amazon EBS Volumes (p. 660)
- Elastic Load Balancers (p. 662)
- AWS Elastic Beanstalk Applications (p. 664)
- AWS CloudFormation Stacks (p. 666)
- Amazon Performance Views (p. 668)
- Amazon CloudWatch Metric Alarms (p. 669)
- AWS Alerts (p. 670)
- Watcher Nodes (System Center Operations Manager 2007 R2) (p. 671)

EC2 Instances

View the health state of the EC2 instances for a particular AWS account, from all Availability Zones and regions. The view also includes EC2 instances running in a virtual private cloud (VPC). The AWS Management Pack retrieves tags, so you can search and filter the list using those tags.

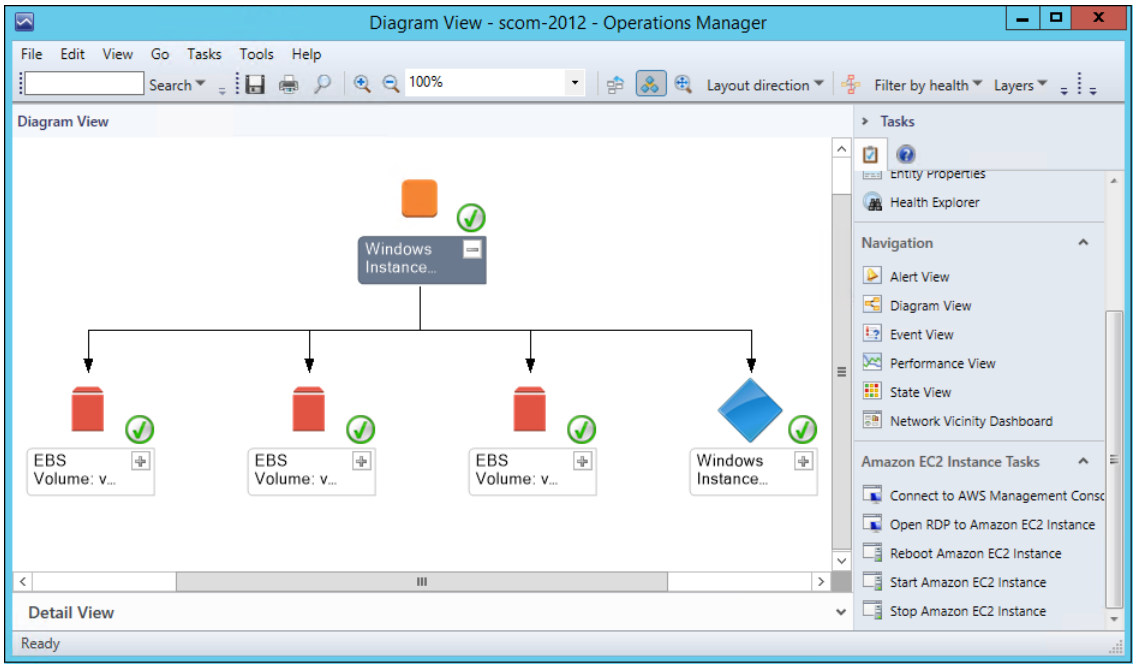


When you select an EC2 instance, you can perform instance health tasks:

- **Open Amazon Console**: Launches the AWS Management Console in a web browser.
- **Open RDP to Amazon EC2 Instance**: Opens an RDP connection to the selected Windows instance.
- **Reboot Amazon EC2 Instance**: Reboots the selected EC2 instance.
- **Start Amazon EC2 Instance**: Starts the selected EC2 instance.
- **Stop Amazon EC2 Instance**: Stops the selected EC2 instance.

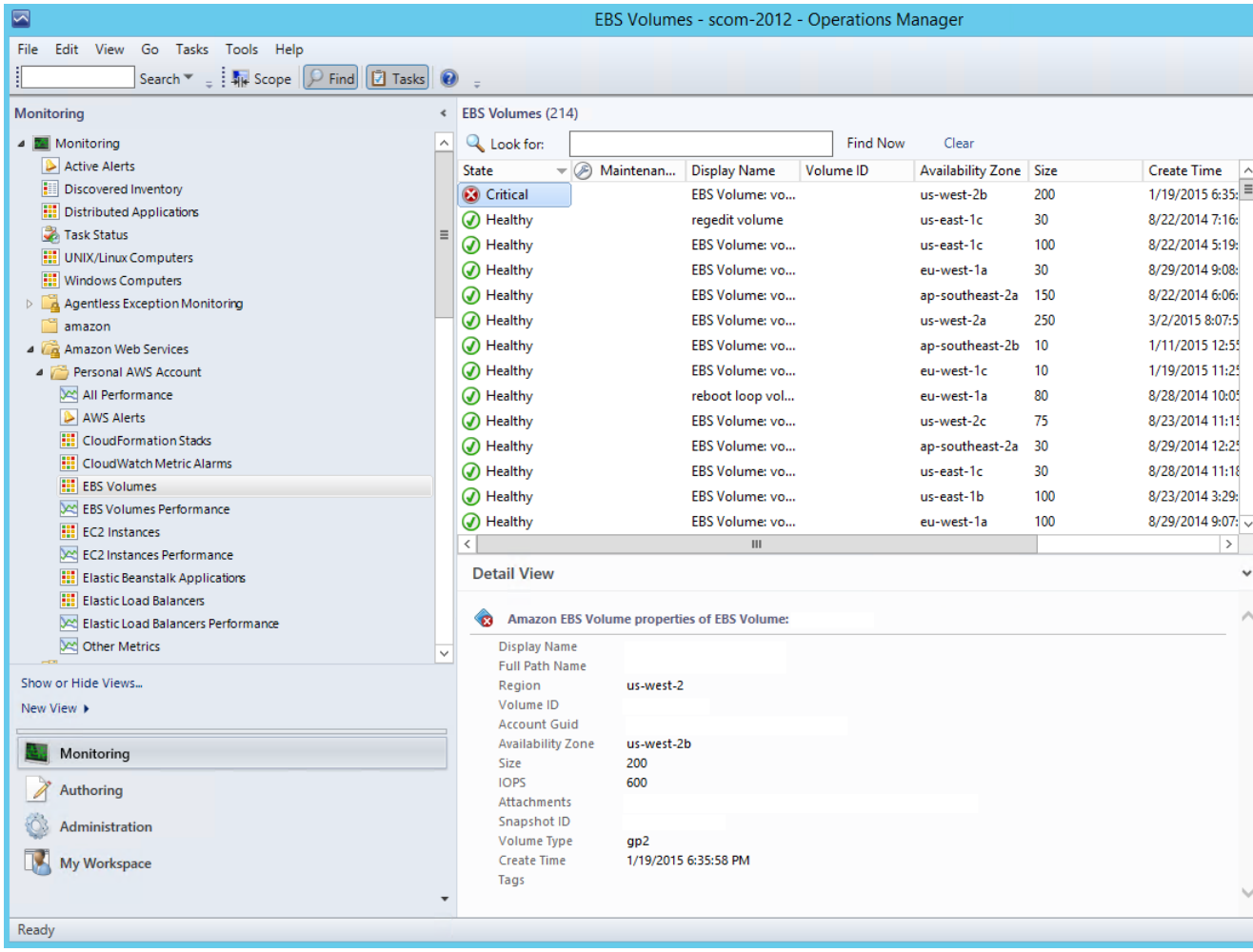
EC2 Instances Diagram View

Shows the relationship of an instance with other components.



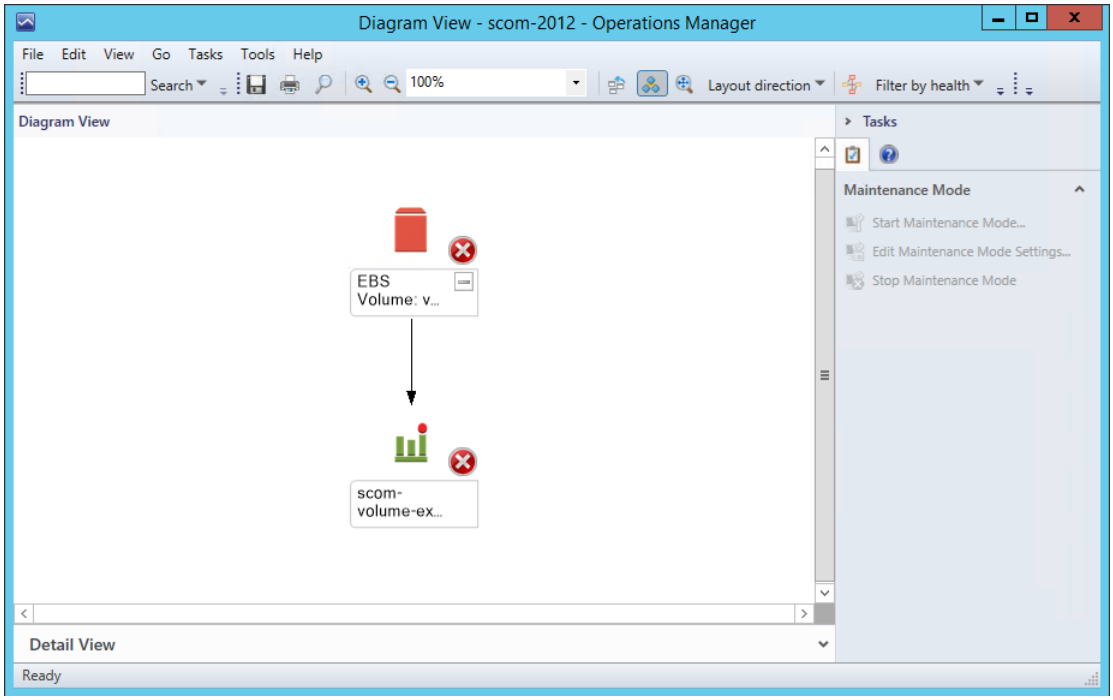
Amazon EBS Volumes

Shows the health state of all the Amazon EBS volumes for a particular AWS account from all Availability Zones and regions.



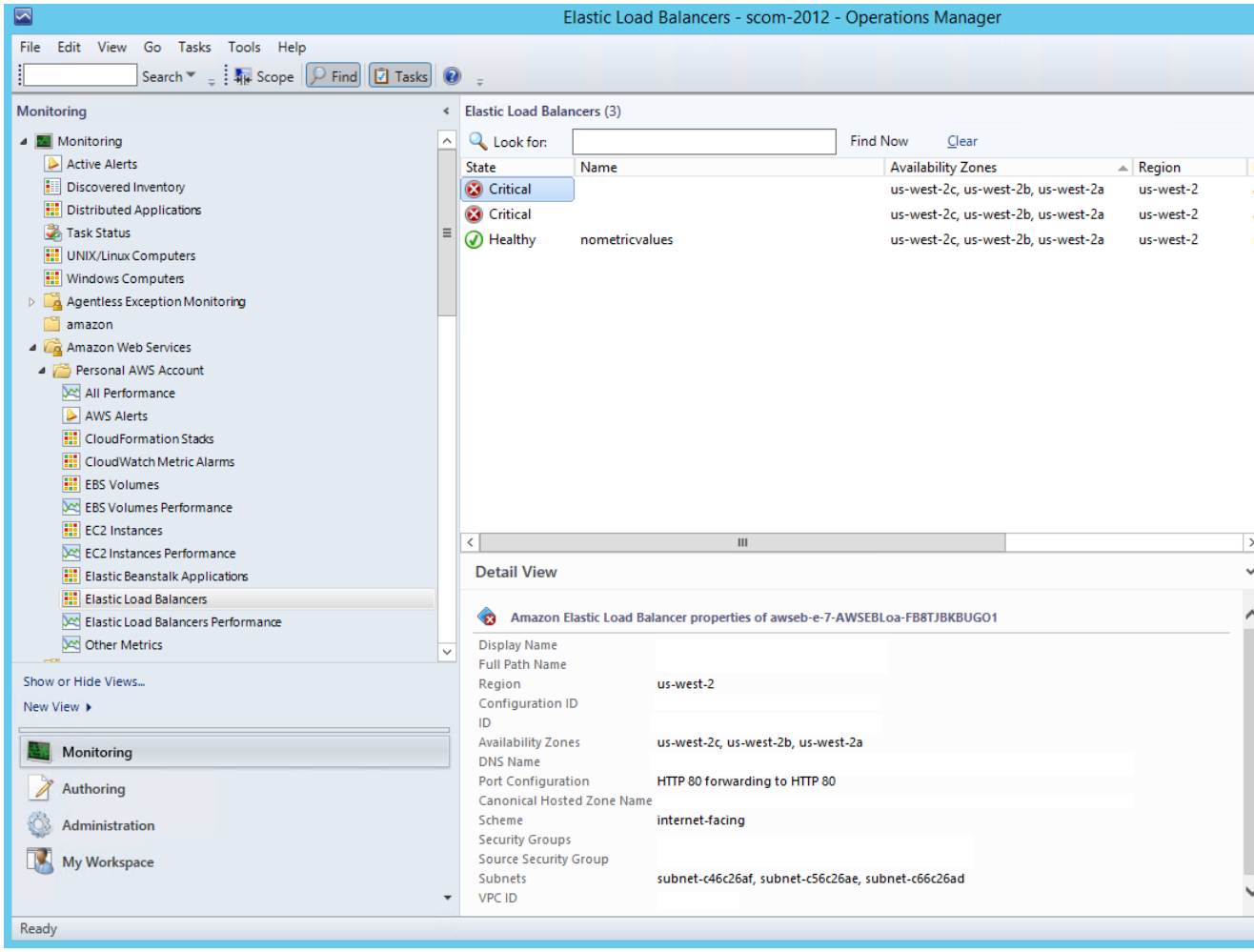
Amazon EBS Volumes Diagram View

Shows an Amazon EBS volume and any associated alarms. The following illustration shows an example:



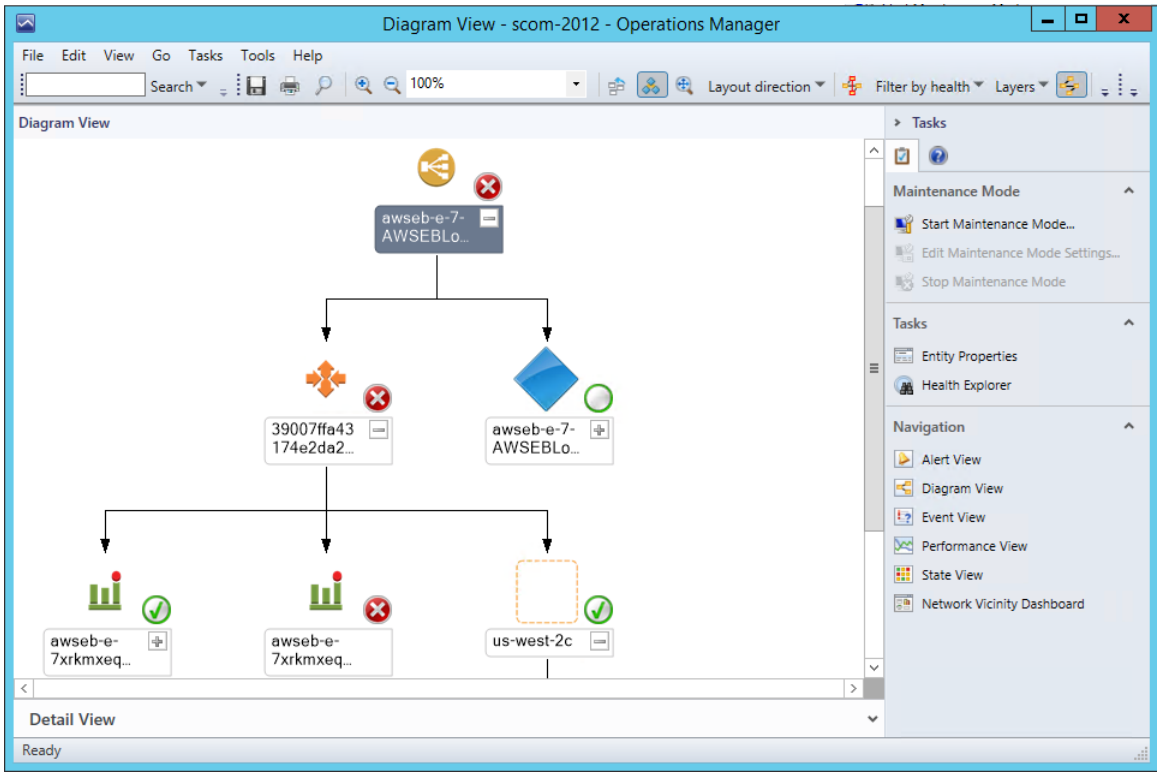
Elastic Load Balancers

Shows the health state of all the load balancers for a particular AWS account from all regions.



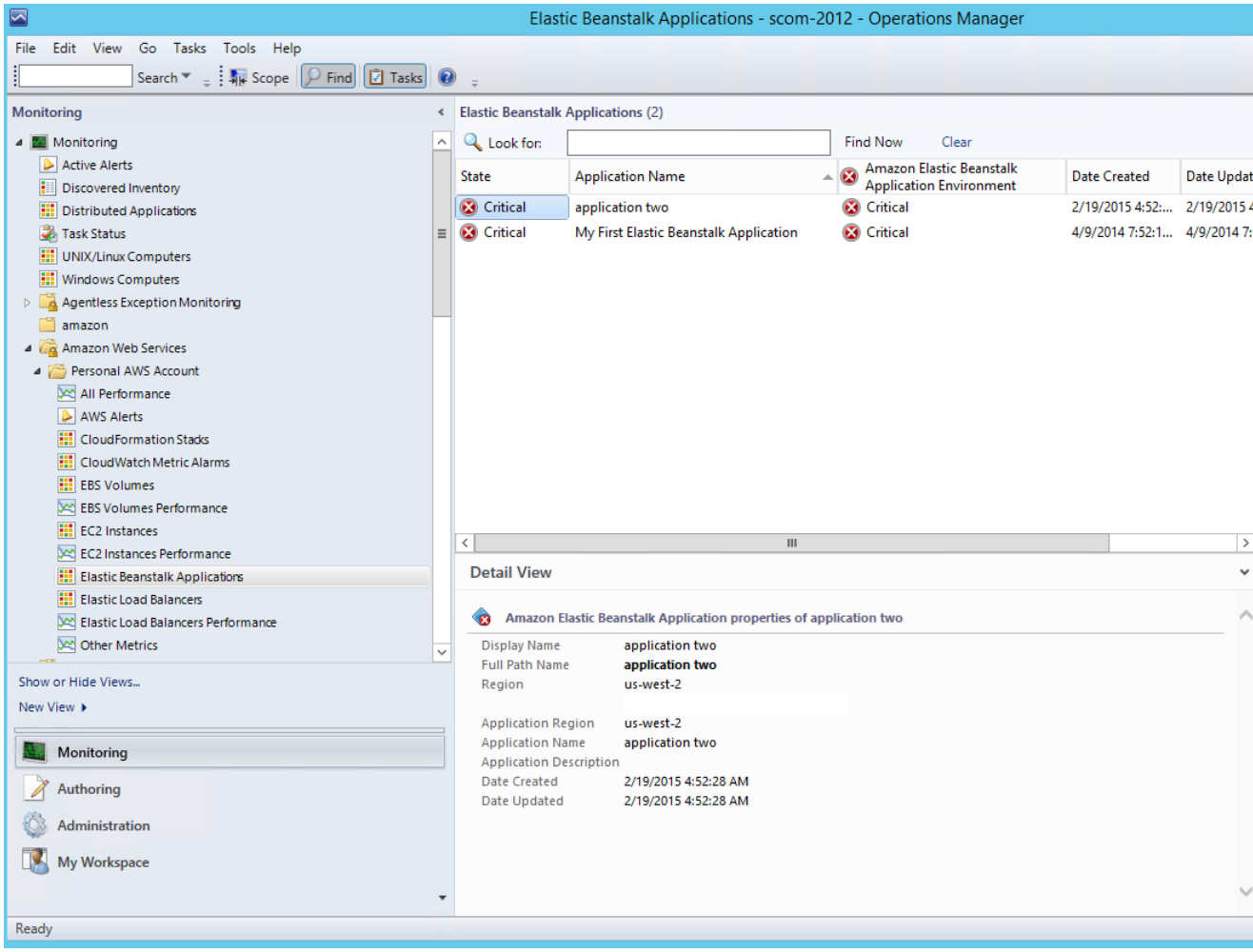
Elastic Load Balancing Diagram View

Shows the Elastic Load Balancing relationship with other components. The following illustration shows an example:



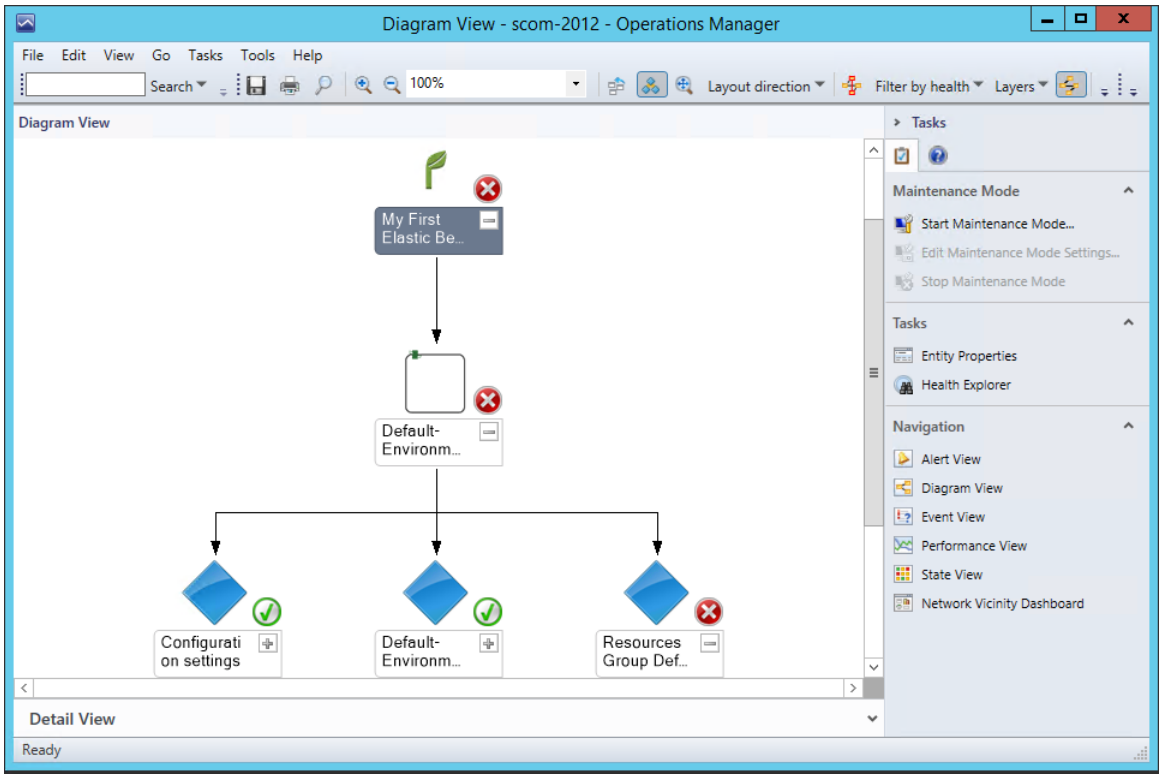
AWS Elastic Beanstalk Applications

Shows the state of all discovered AWS Elastic Beanstalk applications.



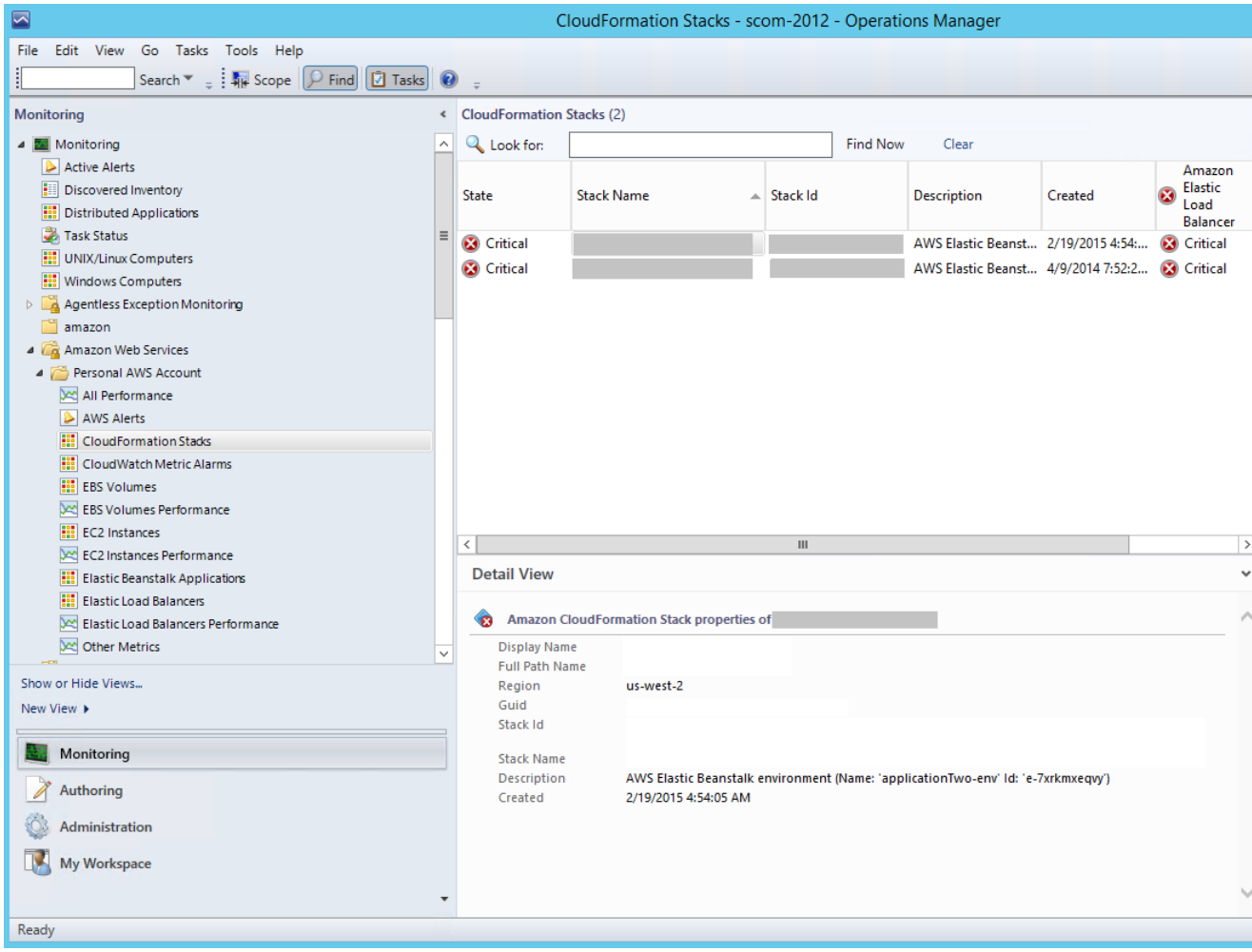
AWS Elastic Beanstalk Applications Diagram View

Shows the AWS Elastic Beanstalk application, application environment, application configuration, and application resources objects.



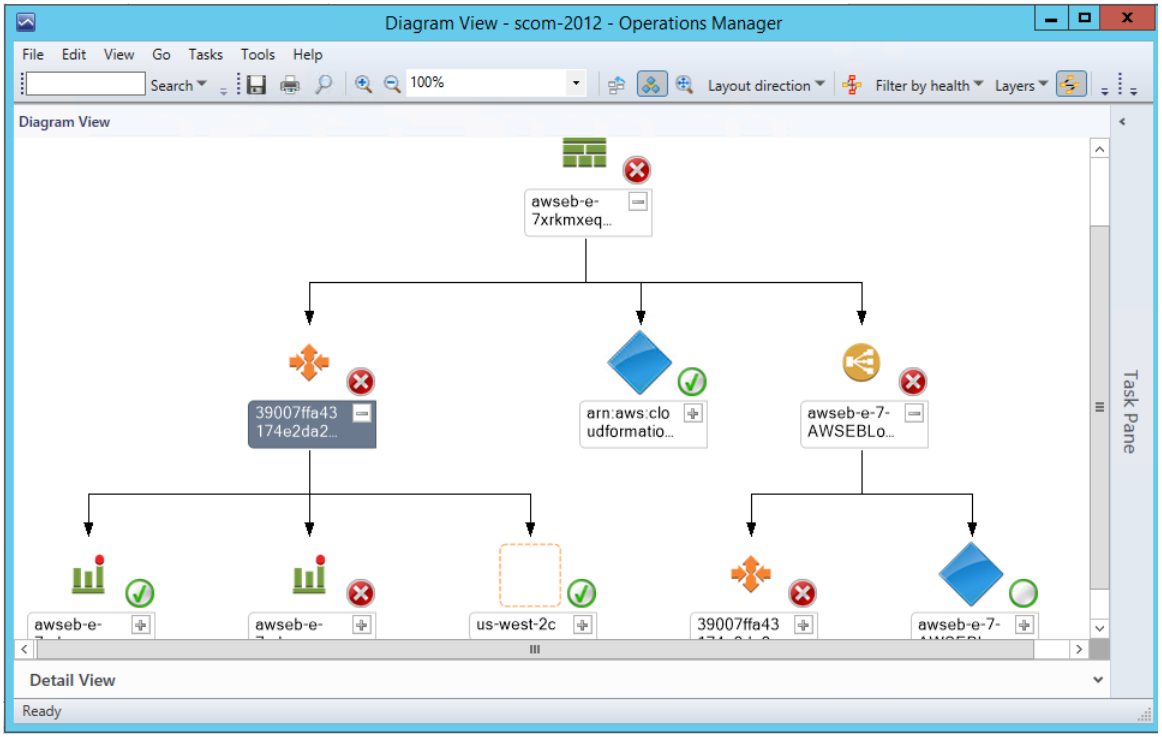
AWS CloudFormation Stacks

Shows the health state of all the AWS CloudFormation stacks for a particular AWS account from all regions.



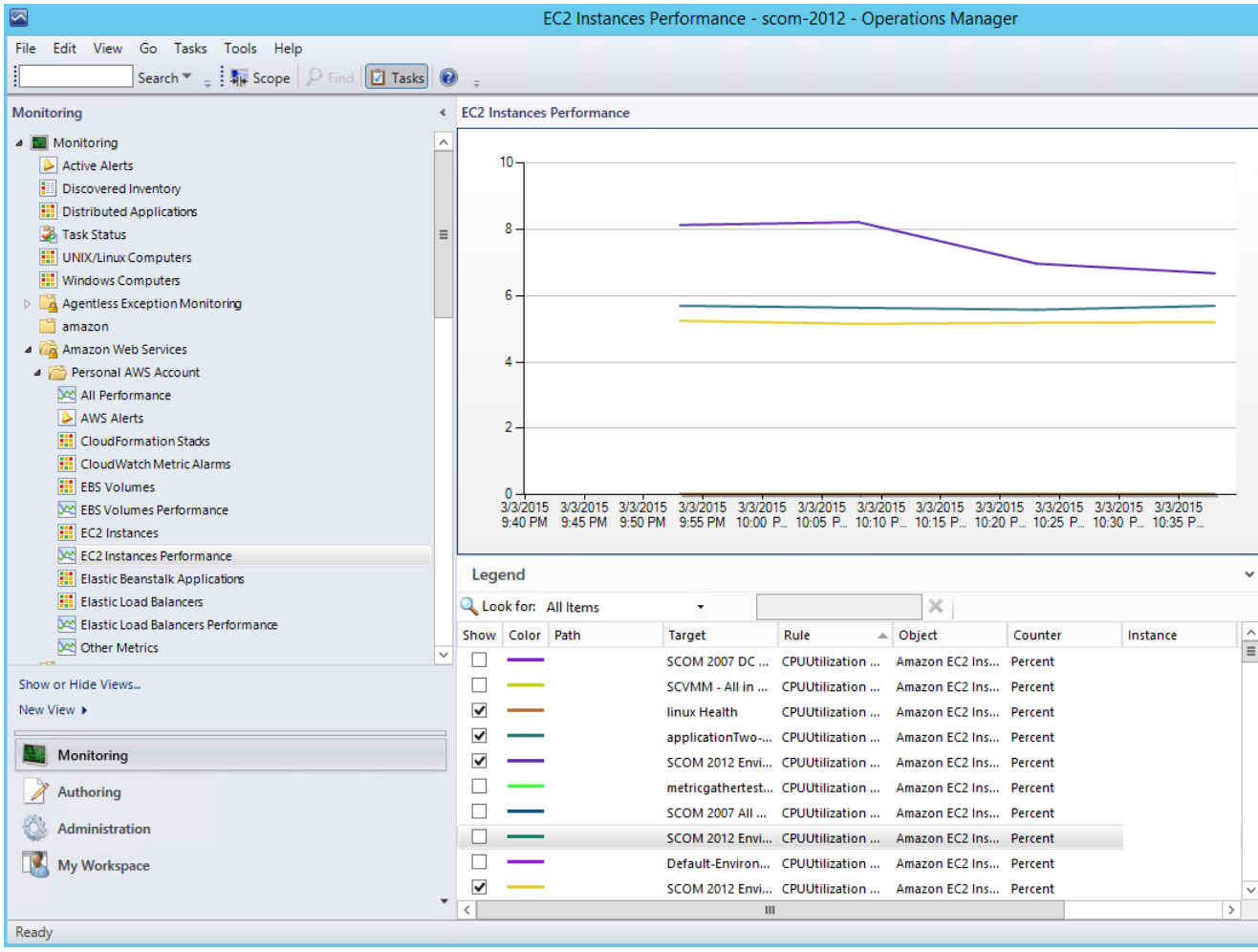
AWS CloudFormation Stacks Diagram View

Shows the AWS CloudFormation stack relationship with other components. An AWS CloudFormation stack might contain Amazon EC2 or Elastic Load Balancing resources. The following illustration shows an example:



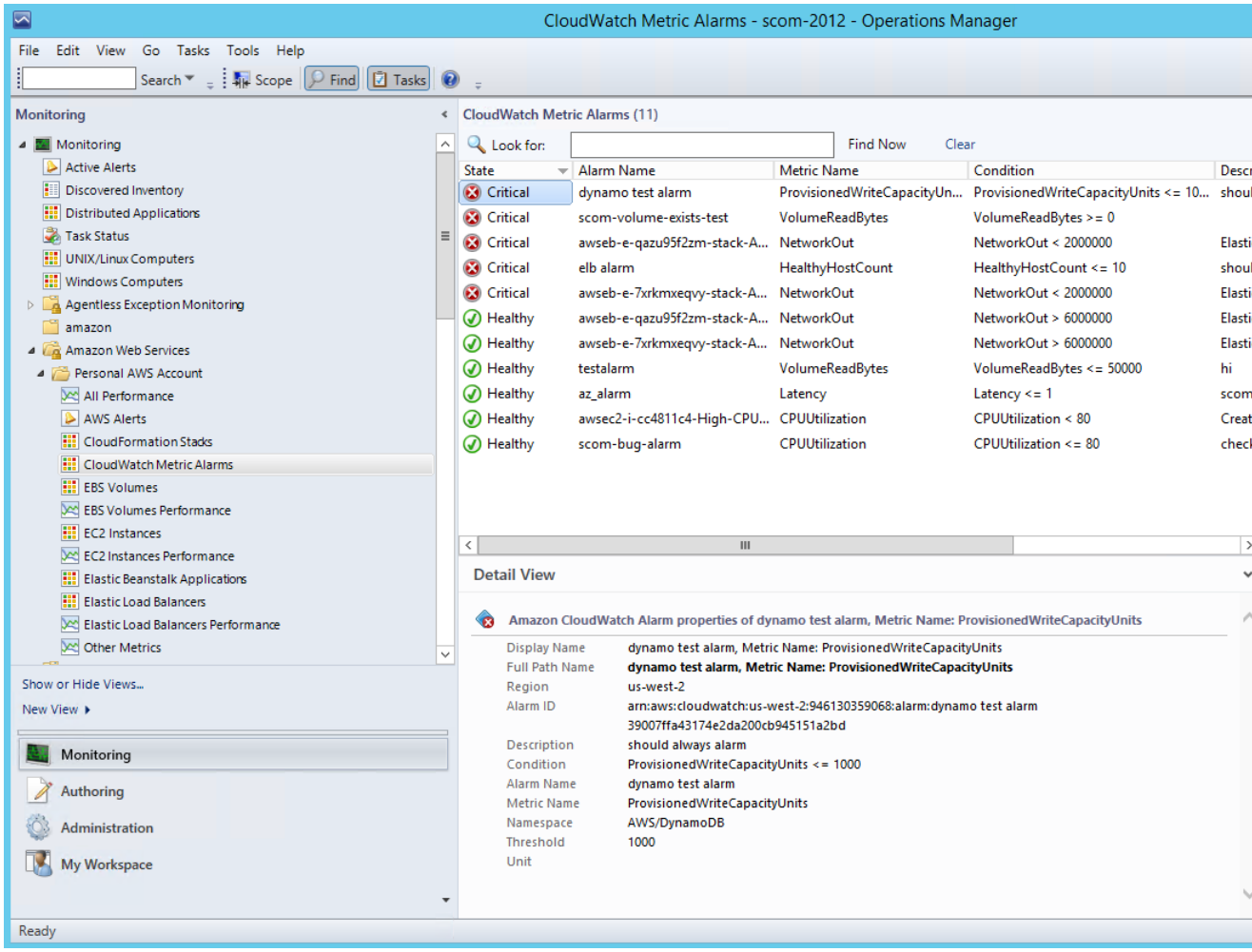
Amazon Performance Views

Shows the Amazon CloudWatch metrics for Amazon EC2, Amazon EBS, and Elastic Load Balancing, custom metrics, and metrics created from CloudWatch alarms. In addition, there are separate performance views for each resource. The **Other Metrics** performance view contains custom metrics, and metrics created from CloudWatch alarms. For more information about these metrics, see the CloudWatch Metrics, Namespaces, and Dimensions Reference in the *Amazon CloudWatch Developer Guide*. The following illustration shows an example.



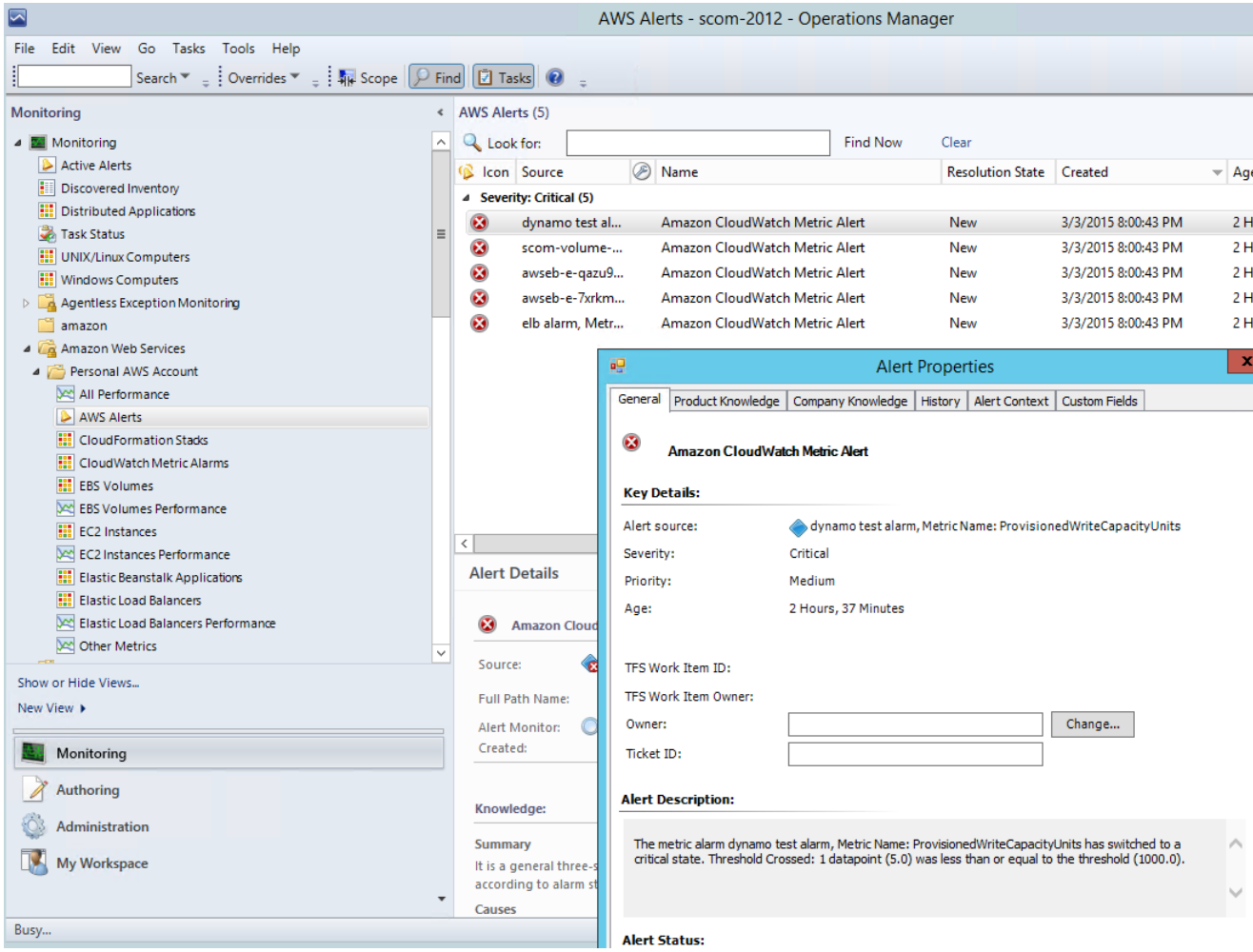
Amazon CloudWatch Metric Alarms

Shows Amazon CloudWatch alarms related to the discovered AWS resources.



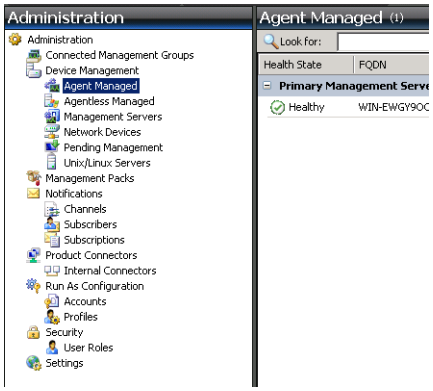
AWS Alerts

Shows the alerts that the AWS management pack produces when the health of an object is in a critical state.



Watcher Nodes (System Center Operations Manager 2007 R2)

View the health state of the watcher nodes across all of the AWS accounts that are being monitored. A **Healthy** state means that the watcher node is configured correctly and can communicate with AWS.



Discoveries

Discoveries are the AWS resources that are monitored by the AWS Management Pack. The AWS Management Pack discovers the following objects:

- Amazon EC2 instances
- EBS volumes
- ELB load balancers
- AWS CloudFormation stacks
- Amazon CloudWatch alarms
- AWS Elastic Beanstalk applications
- Auto Scaling groups and Availability Zones

Amazon CloudWatch metrics are generated for the following resources:

- Amazon EC2 instance
- EBS volume
- Elastic Load Balancing
- Custom Amazon CloudWatch metrics
- Metrics from existing Amazon CloudWatch alarms

For Amazon CloudWatch metrics discovery, the following guidelines apply:

- AWS CloudFormation stacks do not have any default Amazon CloudWatch metrics.
- Stopped Amazon EC2 instances or unused Amazon EBS volumes do not generate data for their default Amazon CloudWatch metrics.
- After starting an Amazon EC2 instance, it can take up to 30 minutes for the Amazon CloudWatch metrics to appear in Operations Manager.
- Amazon CloudWatch retains the monitoring data for two weeks, even if your AWS resources have been terminated. This data appears in Operations Manager.
- An existing Amazon CloudWatch alarm for a resource that is not supported will create a metric and be associated with the Amazon CloudWatch alarm. These metric can be viewed in the Other Metrics performance view.

The AWS Management Pack also discovers the following relationships:

- AWS CloudFormation stack and its Elastic Load Balancing or Amazon EC2 resources
- Elastic Load Balancing load balancer and its EC2 instances
- Amazon EC2 instance and its EBS volumes
- Amazon EC2 instance and its operating system
- AWS Elastic Beanstalk application and its environment, configuration, and resources

The AWS Management Pack automatically discovers the relationship between an EC2 instance and the operating system running on it. To discover this relationship, the Operations Manager Agent must be installed and configured on the instance and the corresponding operating system management pack must be imported in Operations Manager.

Discoveries run on the management servers in the resource pool (System Center 2012) or the watcher node (System Center 2007 R2).

Discovery	Interval (seconds)
Amazon Resources Discovery (SCOM 2012) Discovers EC2 instances, Amazon EBS volumes, load balancers, and CloudFront stacks.	14400
AWS Elastic Beanstalk Discovery Discovers AWS Elastic Beanstalk and its relationship with environment, resources, and configuration.	14400
CloudWatch Alarms Discovery Discovers alarms generated using CloudWatch metrics.	900
Custom CloudWatch Metric Discovery Discovers custom CloudWatch metrics.	14400
Watcher Node Discovery (SCOM 2007 R2) Targets the root management server and creates the watcher node objects.	14400

Monitors

Monitors are used to measure the health of your AWS resources. Monitors run on the management servers in the resource pool (System Center 2012) or the watcher node (System Center 2007 R2).

Monitor	Interval (seconds)
AWS CloudFormation Stack Status	900
Amazon CloudWatch Metric Alarm	300
Amazon EBS Volume Status	900
Amazon EC2 Instance Status	900
Amazon EC2 Instance System Status	900
AWS Elastic Beanstalk Status	900
Watcher Node to Amazon Cloud Connectivity (SCOM 2007 R2)	900

Rules

Rules create alerts (based on Amazon CloudWatch metrics) and collect data for analysis and reporting.

Rule	Interval (seconds)
AWS Resource Discovery Rule (SCOM 2007 R2) Targets the watcher node and uses the AWS API to discover objects for the following AWS resources: EC2 instances, EBS volumes, load balancers, and AWS CloudFormation stacks. (CloudWatch metrics or alarms are not discovered). After discovery is complete, view the objects in the **Not Monitored** state.	14400
Amazon Elastic Block Store Volume Performance Metrics Data Collection Rule	900
Amazon EC2 Instance Performance Metrics Data Collection Rule	900
Elastic Load Balancing Balancing Performance Metrics Data Collection Rule	900
Custom CloudWatch Metric Data Collection Rule	900

Events

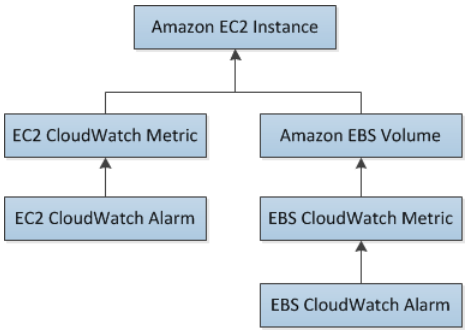
Events report on activities that involve the monitored resources. Events are written to the Operations Manager event log.

Event ID	Description
4101	Amazon EC2 Instance Discovery (General Discovery) finished
4102	Elastic Load Balancing Metrics Discovery, Amazon EBS Volume Metrics Discovery, Amazon EC2 Instance Metrics Discovery finished
4103	Amazon CloudWatch Metric Alarms Discovery finished
4104	Amazon Windows Computer Discovery finished
4105	Collecting Amazon Metrics Alarm finished
4106	EC2 Instance Computer Relation Discovery finished
4107	Collecting AWS CloudFormation Stack State finished
4108	Collecting Watcher Node Availability State finished
4109	Amazon Metrics Collection Rule finished
4110	Task to change Amazon Instance State finished
4111	EC2 Instance Status Monitor State finished
4112	Amazon EBS Volume Status Monitor State finished
4113	Amazon EC2 Instance Scheduled Events Monitor State calculated
4114	Amazon EBS Scheduled Events Monitor State calculated
4115	Elastic Beanstalk Discovery finished
4116	Elastic Beanstalk Environment Status State calculated

Event ID	Description
4117	Elastic Beanstalk Environment Operational State calculated
4118	Elastic Beanstalk Environment Configuration State calculated

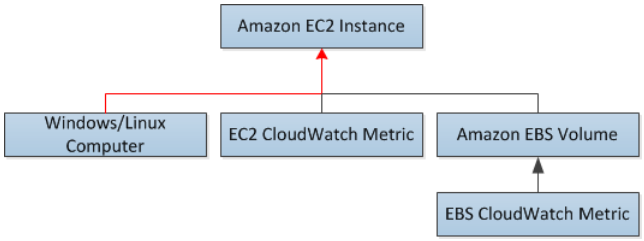
Health Model

The following illustration shows the health model defined by the AWS Management Pack.



The health state for a CloudWatch alarm is rolled up to its corresponding CloudWatch metric. The health state for a CloudWatch metric for Amazon EC2 is rolled up to the EC2 instance. Similarly, the health state for the CloudWatch metrics for Amazon EBS is rolled up to the Amazon EBS volume. The health states for the Amazon EBS volumes used by an EC2 instance are rolled up to the EC2 instance.

When the relationship between an EC2 instance and its operating system has been discovered, the operating system health state is rolled up to the EC2 instance.



The health state of an AWS CloudFormation stack depends on the status of the AWS CloudFormation stack itself and the health states of its resources, namely the load balancers and EC2 instances.

The following table illustrates how the status of the AWS CloudFormation stack corresponds to its health state.

Health State	AWS CloudFormation Stack Status	Notes

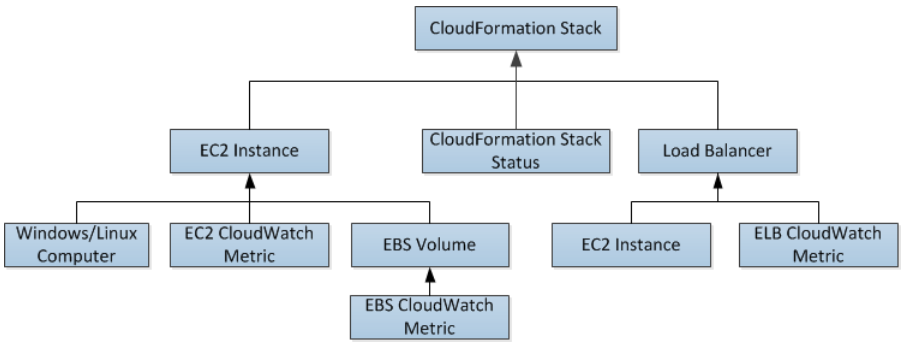
Error	`CREATE_FAILED`
	`DELETE_IN_PROGRESS`
	`DELETE_FAILED`
	`UPDATE_ROLLBACK_FAILED`
		Most likely usable

Health State	AWS CloudFormation Stack Status	Notes

Warning	`UPDATE_ROLLBACK_IN_PROGRESS`
	
	`UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS`
	
	`UPDATE_ROLLBACK_COMPLETE`
		Recovering after some problem

Healthy	`CREATE_COMPLETE`
	
	`UPDATE_IN_PROGRESS`
	
	`UPDATE_COMPLETE_CLEANUP_IN_PROGRESS`
	
	`UPDATE_COMPLETE`
		Usable

The full health model for an AWS CloudFormation stack is as follows:



Customizing the AWS Management Pack

To change the frequency of discoveries, rules, and monitors, you can override the interval time (in seconds).

To change frequency

1. In the **Operations Manager** toolbar, click **Go**, and then click **Authoring**.
2. In the **Authoring** pane, expand **Management Pack Objects** and then click the object to change (for example, **Object Discoveries**, **Rules**, or **Monitors**).
3. In the toolbar, click **Scope**.
4. In the **Scope Management Pack Objects** dialog box, click **View all targets**.
5. To limit the scope to Amazon objects, type Amazon in the **Look for** field.
6. Select the object want to configure and click **OK**.
7. In the **Operations Manager** center pane, right-click the object to configure, click **Overrides**, and then click the type of override you want to configure.
8. Use the **Override Properties** dialog box to configure different values and settings for objects.

Tip
To disable a discovery, rule, or monitoring object right-click the object to disable in the **Operations Manager** center pane, click **Overrides**, and then click **Disable the Rule**. You might disable rules if, for example, you do not run AWS Elastic Beanstalk applications or use custom Amazon CloudWatch metrics.

For information about creating overrides, see Tuning Monitoring by Using Targeting and Overrides on the *Microsoft TechNet* website.

For information about creating custom rules and monitors, see Authoring for System Center 2012 - Operations Manager or System Center Operations Manager 2007 R2 Management Pack Authoring Guide on the *Microsoft TechNet* website.

Upgrading the AWS Management Pack

The procedure that you'll use to update AWS Management Pack depends on the version of System Center.

System Center 2012

To upgrade the AWS Management Pack

1. On the AWS Add-Ins for Microsoft System Center website, click **SCOM 2012**. Download `AWS-SCOM-MP-2.0-2.5.zip` to your computer and unzip it. The `.zip` file includes `Amazon.AmazonWebServices.mpb`.
2. In the Operations console, on the **Go** menu, click **Administration**, and then click **Management Packs**.
3. In the **Tasks** pane, click **Import Management Packs**.
4. On the **Select Management Packs** page, click **Add**, and then click **Add from disk**.
5. In the **Select Management Packs to import** dialog box, select the `Amazon.AmazonWebServices.mpb` file from the location where you downloaded it, and then click **Open**.
6. On the **Select Management Packs** page, under **Import list**, select the **Amazon Web Services** management pack, and then click **Install**.

   If the **Install** button is disabled, upgrading to the current version is not supported and you must uninstall the AWS Management Pack before you can install the current version. For more information, see Uninstalling the AWS Management Pack (p. 678).

System Center 2007 R2

To upgrade the AWS Management Pack

1. On the Management Server, go to the AWS Add-Ins for Microsoft System Center website and click **SCOM 2007**. Save `AWS-MP-Setup-2.5.msi`, and then run it.
2. Click **Next** and follow the directions to upgrade the components that you installed previously.
3. If your root management server, Operations console, and watcher node are on different computers, you must download and run the setup program on each computer.
4. On the watcher node, open a Command Prompt window as an administrator and run the following commands.

```
C:\> net stop HealthService
The System Center Management service is stopping.
The System Center Management service was stopped successfully.

C:\> net start HealthService
```

```
The System Center Management service is starting.
The System Center Management service was started successfully.
```

5. In the Operations console, on the **Go** menu, click **Administration**, and then click **Management Packs**.
6. In the **Actions** pane, click **Import Management Packs**.
7. On the **Select Management Packs** page, click **Add**, and then click **Add from disk**.
8. In the **Select Management Packs to import** dialog box, change the directory to `C:\Program Files (x86)\Amazon Web Services Management Pack`, select the `Amazon.AmazonWebServices.mp` file, and then click **Open**.
9. On the **Select Management Packs** page, under **Import list**, select the **Amazon Web Services** management pack, and then click **Install**.

   If the **Install** button is disabled, upgrading to the current version is not supported and you must uninstall AWS Management Pack first. For more information, see Uninstalling the AWS Management Pack (p. 678).

Uninstalling the AWS Management Pack

If you need to uninstall the AWS Management Pack, use the following procedure.

System Center 2012

To uninstall the AWS Management Pack

1. In the Operations console, on the **Go** menu, click **Administration**, and then click **Management Packs**.
2. Right-click **Amazon Web Services** and select **Delete**.
3. In the **Dependent Management Packs** dialog box, note the dependent management packs, and then click **Close**.
4. Right-click the dependent management pack and select **Delete**.
5. Right-click **Amazon Web Services** and select **Delete**.

System Center 2007 R2

To uninstall the AWS Management Pack

1. Complete steps 1 through 5 described for System Center 2012 in the previous section.
2. From Control Panel, open Programs and Features. Select `Amazon Web Services Management Pack` and then click **Uninstall**.
3. If your root management server, Operations console, and watcher node are on different computers, you must repeat this process on each computer.

Troubleshooting the AWS Management Pack

The following are common errors, events, and troubleshooting steps.

Contents

- Errors 4101 and 4105 (p. 679)
- Error 4513 (p. 679)
- Event 623 (p. 679)
- Events 2023 and 2120 (p. 680)
- Event 6024 (p. 680)
- General Troubleshooting for System Center 2012 — Operations Manager (p. 680)
- General Troubleshooting for System Center 2007 R2 (p. 681)

Errors 4101 and 4105

If you receive one of the following errors, you must upgrade the AWS Management Pack. For more information, see Upgrading the AWS Management Pack (p. 677).

```
Error 4101
Exception calling "DescribeVolumes" with "1" argument(s): "AWS was not able to
 validate the
provided access credentials"
```

```
Error 4105
Exception calling "DescribeApplications" with "0" argument(s): "The security
token included
in the request is invalid"
```

Error 4513

If you receive one of the following error, you must upgrade the AWS Management Pack. For more information, see Upgrading the AWS Management Pack (p. 677).

```
Error 4513
The callback method DeliverDataToModule failed with exception "Resolution of
the dependency
failed, type = "Amazon.SCOM.SDK.Interfaces.IMonitorSdk", name = "(none)".
Exception occurred while: Calling constructor Amazon.SCOM.SDK.CloudWatch.AwsMon
itorSdk
(System.String awsAccessKey, System.String awsSecretKey).
Exception is: InvalidOperationException - Collection was modified; enumeration
 operation
may not execute.
```

Event 623

If you find the following event in the Windows event log, follow the solution described in KB975057.

```
Event ID: 623
HealthService (process_id) The version store for instance instance ("name") has
 reached
its maximum size of size MB. It is likely that a long-running transaction is
preventing
cleanup of the version store and causing it to build up in size. Updates will
be rejected
```

```
until the long-running transaction has been completely committed or rolled back.

Possible long-running transaction:
SessionId: id
Session-context: value
Session-context ThreadId: id
Cleanup: value
```

Events 2023 and 2120

If you find the following events in the Windows event log, see Event ID 2023 and 2120 for more information.

```
Event ID: 2023
The Health Service has removed some items from the send queue for management
group "Servers"
since it exceeded the maximum allowed size of size megabytes.
```

```
Event ID: 2120
The Health Service has deleted one or more items for management group "Servers"
 which could
not be sent in 1440 minutes.
```

Event 6024

If you find the following event in the Windows event log, see Health Service Restarts for more information.

```
Event ID: 6024
LaunchRestartHealthService.js : Launching Restart Health Service. Health Service
 exceeded
Process\Handle Count or Private Bytes threshold.
```

General Troubleshooting for System Center 2012 — Operations Manager

Try the following to resolve any issues.

- Verify that you have installed the latest Update Rollup for System Center 2012 — Operations Manager. The AWS Management Pack requires at least Update Rollup 1.
- Ensure that you have configured the AWS Management Pack after importing it by running the Add Monitoring Wizard. For more information, see Step 1: Installing the AWS Management Pack (p. 649).
- Verify that you have waited long enough for the AWS resources to be discovered (10–20 minutes).
- Verify that the management servers are configured properly.
  - Management servers must have Internet connectivity.
  - The action account for a management server must have local administrator privileges on the management server.
  - The management server must have the .NET Framework 4.5. or later.
- Verify that the AWS Run As account is valid.
  - The values for the access key ID and secret access key are correct.

- The access keys are active: In the AWS Management Console, click your name in the navigation bar and then click **Security Credentials**.
- The IAM user has at least read-only access permission. Note that read-only access allows the user actions that do not change the state of a resource, such as monitoring, but do not allow the user actions like launching or stopping an instance.
  - If an Amazon CloudWatch metric shows as **Not Monitored**, check whether at least one Amazon CloudWatch alarm has been defined for that Amazon CloudWatch metric.
  - For further troubleshooting, use the information in the event logs.
  - Check the Operations Manager event log on the management server. For more information, see Events (p. 674) for a list of the events that the AWS Management Pack writes to the Operations Manager event log.

General Troubleshooting for System Center 2007 R2

Try the following to resolve any issues.

- Ensure that you have configured the AWS Management Pack after importing it by running the Add Monitoring Wizard. For more information, see Step 1: Installing the AWS Management Pack (p. 649).
- Verify that you have waited long enough for the AWS resources to be discovered (10–20 minutes).
- Verify that the watcher node is configured properly.
  - The proxy agent is enabled. For more information, see Step 2: Configuring the Watcher Node (p. 650).
  - The watcher node has Internet connectivity.
  - The action account for the watcher node has local administrator privileges.
  - The watcher node must have the .NET Framework 3.5.1 or later.
- Verify that the watcher node is healthy and resolve all alerts. For more information, see Views (p. 658).
- Verify that the AWS Run As account is valid.
  - The values for the access key ID and secret access key are correct.
  - The access keys are active: In the AWS Management Console, click your name in the navigation bar and then click **Security Credentials**.
  - The IAM user has at least read-only access permission. Note that read-only access allows the user actions that do not change the state of a resource, such as monitoring, but do not allow the user actions like launching or stopping an instance.
    - If an Amazon CloudWatch metric shows as **Not Monitored**, check whether at least one Amazon CloudWatch alarm has been defined for that Amazon CloudWatch metric.
    - For further troubleshooting, use the information in the event logs.
    - Check the Operations Manager event log on the management server as well as the watcher node. For more information, see Events (p. 674) for a list of the events that the AWS Management Pack writes to the Operations Manager event log.

AWS Diagnostics for Microsoft Windows Server - Beta

AWS Diagnostics for Microsoft Windows Server is a easy-to-use tool that you run on an Amazon EC2 Windows Server instance to diagnose and troubleshoot possible problems. It is valuable not just for collecting log files and troubleshooting issues, but also proactively searching for possible areas of concern. For example, this tool can diagnose configuration issues between the Windows Firewall and the AWS security groups that might affect your applications. It can even examine EBS boot volumes from other instances and collect relevant logs for troubleshooting Windows Server instances using that volume.

One use for AWS Diagnostics for Microsoft Windows Server is diagnosing problems with Key Management Service (KMS) activations. KMS activation can fail if you have changed the DNS server, added instances to a domain, or if the server time is out of sync. In this case, instead of trying to examine your configuration settings manually and debugging the issue, run the AWS Diagnostics for Microsoft Windows Server tool to give you the information you need about possible issues.

The tool can also find differences between the rules in an security group and the Windows Firewall. If you provide your AWS user credentials to describe your security groups, the AWS Diagnostics for Microsoft Windows Server tool is able verify whether the ports listed in a security group are allowed through the Windows Firewall. You eliminate the need to look at firewall rules manually and verify them against the security group rules.

The AWS Diagnostics for Microsoft Windows Server tool is free and can be downloaded and installed from AWS Diagnostics for Microsoft Windows Server - Beta.

AWS Diagnostics for Microsoft Windows Server has two different modules: a data collector module that collects data from all different sources, and an analyzer module that parses the data collected against a series of predefined rules to identify issues and provide suggestions.

The AWS Diagnostics for Microsoft Windows Server tool only runs on Windows Server running on an EC2 instance. When the tool starts, it checks whether it is running on an EC2 instance. If the check fails, the tool displays the `EC2InstanceCheckFailed` error message.

Analysis Rules

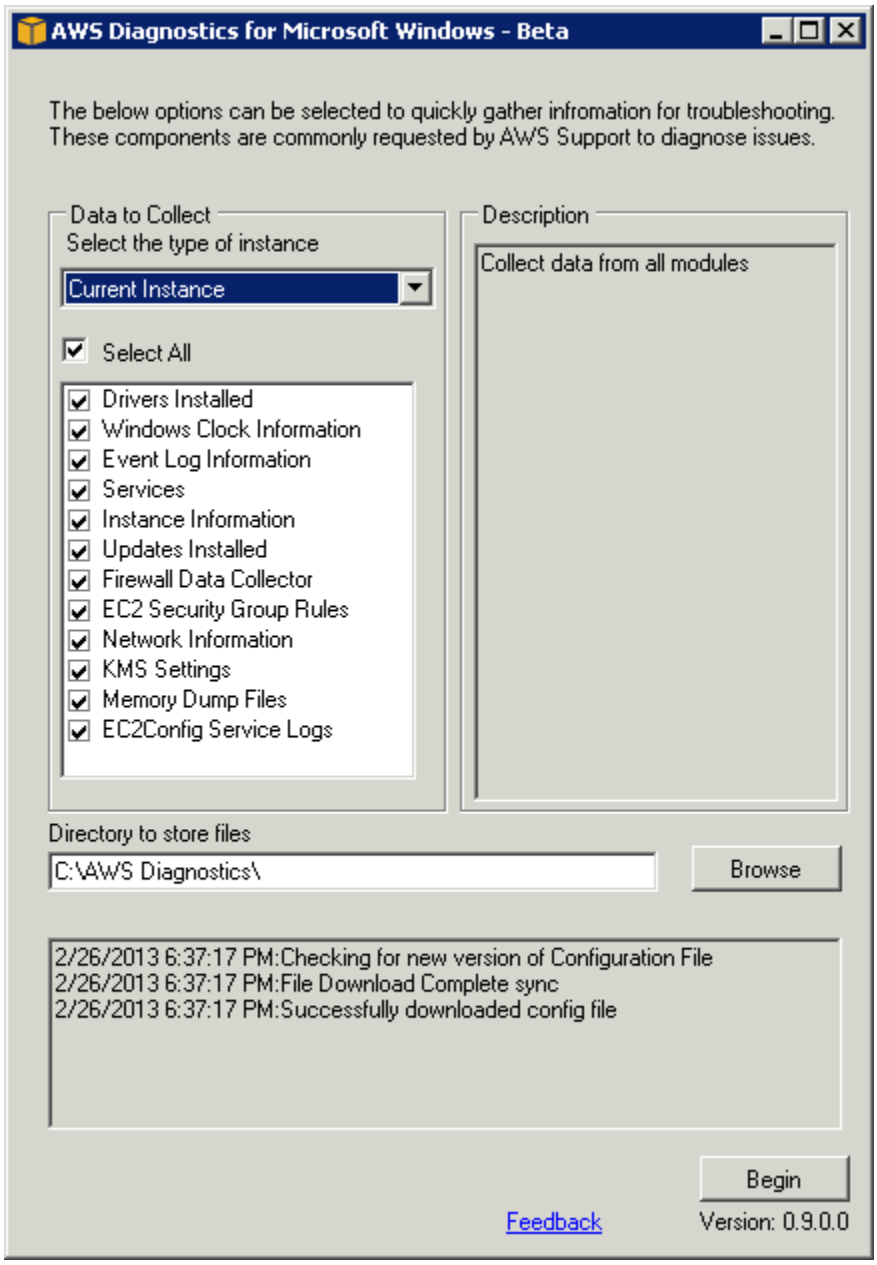
AWS Diagnostics for Microsoft Windows Server provides the following analysis rules:

- Check for activation status and KMS settings
- Check for proper route table entries for metadata and KMS access
- Compare security group rules with Windows Firewall rules
- Check the version of the PV driver (RedHat or Citrix)
- Check whether the `RealTimeIsUniversal` registry key is set
- Check the default gateway settings if using multiple NICs
- Bug check code in mini dump files

Even if the analyzer doesn't report any problems, the data collected by the tool might still be useful. You can view the data files created by the tool to look for problems or provide these files to AWS Support to help resolve a support case.

Analyzing the Current Instance

To analyze the current instance, run the AWS Diagnostics for Microsoft Windows Server tool and select **Current Instance** for the type of instance. In the **Data to Collect** section of the main window, specify the data that AWS Diagnostics for Microsoft Windows Server collects.



Data	Description
Drivers Installed	Collects information about all drivers installed on the instance.
Windows Clock Information	Collects current time and time zone information for the instance.
Event Log Information	Collects critical, error, and warning messages from the event logs.
Services	Collects information about the services that are installed on the instance.

Data	Description
Instance Information	Collects information from the instance metadata and local environment variables.
Updates Installed	Collects information about the updates that are installed on the instance.
Firewall Data Collector	Collects information about the Windows Firewall settings.
EC2 Security Group Rules	Collects information about the rules in the Amazon EC2 security groups associated with the instance.
Network Information	Collects route table and IP address information for the instance.
KMS Settings	Collects Key Management Service settings.
Memory Dump Files	Collects any memory dump files that exist on the instance.
EC2Config Service Logs	Collects log files generated by the EC2Config service.

Collecting Data From an Offline Instance

The **Offline Instance** option is useful when you want to debug a problem with a Windows instance that is either unable to boot up or is preventing you from running the AWS Diagnostics for Microsoft Windows Server tool on it. In this case, you can detach the EBS boot volume from that instance and attach it to another Windows instance.

To collect data from an offline instance

1. Stop the faulty instance, if it is not stopped already.
2. Detach the EBS boot volume from the faulty instance.
3. Attach the EBS boot volume to another working Windows instance that has AWS Diagnostics for Microsoft Windows Server installed on it
4. Mount the volume in the working instance, assigning it a drive letter (for example, `F:`).
5. Run the AWS Diagnostics for Microsoft Windows Server tool on the working instance and select **Offline Instance**.
6. Choose the drive letter of the newly mounted volume (for example, `F:`).
7. Click **Begin**.

The AWS Diagnostics for Microsoft Windows Server tool scans the volume and collects troubleshooting information based on the log files that are on the volume. For offline instances, the data collected is a fixed set, and no analysis of the data is performed.

Data File Storage

By default, the AWS Diagnostics for Microsoft Windows Server tool places its data files in the directory from which you launch the tool. You can choose where to save the data files that are collected by the AWS Diagnostics for Microsoft Windows Server tool. Within the chosen directory, the tool creates a

directory named `DataCollected`. Each time it runs, the tool also creates a separate directory with the current date and time stamp. Each data collection module produces an XML file that contains information for that data set. Finally, the tool creates a ZIP file archive containing copies of all of the data files generated. You can provide this archive to an AWS support engineer if needed.

Troubleshooting Windows Instances

The following procedures and tips can help you troubleshoot problems with your Amazon EC2 Windows instances.

Common Issues and Messages

- Boot an EC2 Windows Instance into Directory Services Restore Mode (DSRM) (p. 688)
- High CPU shortly after Windows starts (p. 690)
- No console output (p. 690)
- Instance terminates immediately (p. 691)
- "Password is not available" (p. 691)
- "Password not available yet" (p. 692)
- "Cannot retrieve Windows password" (p. 692)
- "Waiting for the metadata service" (p. 693)
- Remote Desktop can't connect to the remote computer (p. 695)
- RDP displays a black screen instead of the desktop (p. 697)
- "Unable to activate Windows" (p. 698)
- "Windows is not genuine (0x80070005)" (p. 699)
- "No Terminal Server License Servers available to provide a license" (p. 699)
- Instance loses network connectivity or scheduled tasks don't run when expected (p. 699)
- Insufficient Instance Capacity (p. 700)
- Instance Limit Exceeded (p. 700)
- Windows Server 2012 R2 not available on the network (p. 700)

If you need additional help, you can post a question to the Amazon EC2 forum. Be sure to post the ID of your instance and any error messages, including error messages available through console output.

To get additional information for troubleshooting problems with your instance, use AWS Diagnostics for Microsoft Windows Server - Beta (p. 682). For information about troubleshooting issues with PV drivers, see Troubleshooting PV Drivers (p. 270).

Boot an EC2 Windows Instance into Directory Services Restore Mode (DSRM)

If an instance running Microsoft Active Directory experiences a system failure or other critical issues you can troubleshoot the instance by booting into a special version of Safe Mode called *Directory Services Restore Mode* (DSRM). In DSRM you can repair or recover Active Directory.

Driver Support for DSRM

How you enable DSRM and boot into the instance depends on the drivers the instance is running. In the EC2 console, you can view driver version details for an instance from the System Log. The following tables shows which drivers are supported for DSRM.

Driver Versions	Operating System	DSRM Supported?	Next Steps
Citrix PV 5.9	Windows Server 2008 or earlier	No	Restore the instance from a backup. You cannot enable DSRM.
AWS PV 7.2.0	Windows Server 2012 R2	No	Though DSRM is not supported for this driver, you can still detach the root volume, attach it to another instance, and enable DSRM (as described in this section).
AWS PV 7.2.2 and later	Windows Server 2012 R2	Yes	Detach the root volume, attach it to another instance, and enable DSRM (as described in this section).
Enhanced Networking (Intel 82599 Virtual Function)	Windows Server 2012 R2	Yes	Detach the root volume, attach it to another instance, and enable DSRM (as described in this section).

Note
By default, Enhanced Networking is enabled on the following Windows Server 2012 R2 instance types:

- C3
- C4
- D2
- I2
- R3

For more information about instance types, see Amazon EC2 Instances. For information about how to enable Enhanced Networking for other Windows Server instances, see Enabling Enhanced Networking on Windows Instances in a VPC.

Configure an Instance to Boot into DSRM

EC2 Windows instances do not have network connectivity before the operating system is running. For this reason, you cannot press the F8 button on your keyboard to select a boot option. You must use one of the following procedures to boot an EC2 Windows Server instance into DSRM.

Boot an Online Instance into DSRM

If you suspect that Active Directory has been corrupted and the instance is still running, you can configure the instance to boot into DSRM using either the System Configuration dialog box or the command prompt. Choose one of the following methods. If your instance is not online (unavailable) see the next section:

To boot an online instance into DSRM using the System Configuration dialog box

1. In the **Run** dialog box type `msconfig` and press Enter.
2. Choose the **Boot** tab.
3. Under **Boot options** choose **Safe boot**.
4. Choose **Active Directory repair** and then choose **OK**. The system prompts you to reboot the server.

To boot an online instance into DSRM using the command prompt

1. Open a command prompt.
2. Type `bcdedit /set safeboot dsrepair` and press Enter.

Boot an Offline Instance into DSRM

If an instance is offline and unreachable you must detach the root volume and attach it to another instance to enable DSRM mode.

To boot an offline instance into DSRM

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Locate the affected instance. Open the context (right-click) menu for the instance, choose **Instance State**, and then choose **Stop**.
4. Choose **Launch Instance** and create a temporary instance in the same Availability Zone as the affected instance. Choose an instance type that uses a different version of Windows. For example, if your instance is Windows Server 2008 R1, then choose a Windows Server 2008 R2 instance.

   Important
   If you do not create the instance in the same Availability Zone as the affected instance you will not be able to attach the root volume of the affected instance to the new instance.
5. In the navigation pane, choose **Volumes**.
6. Locate the root volume of the affected instance. Detach the volume and attach it to the temporary instance you created earlier. Attach it with the default device name (xvdf).
7. Use Remote Desktop to connect to the temporary instance, and then use the Disk Management utility to make the volume available for use.
8. Open a command prompt and run the following command. Replace *D* with the actual drive letter of the secondary volume you just attached:

   ```
   bcdedit /store D:\Boot\BCD /set {default} safeboot dsrepair
   ```
9. In the Disk Management Utility, choose the drive you attached earlier, open the context (right-click) menu, and choose **Offline**.

10. In the EC2 console, detach the affected volume from the temporary instance and reattach it to your original instance with the device name /dev/sda1. You must specify this device name to designate the volume as a root volume.
11. Start the instance.
12. After the instance passes the health checks in the EC2 console, connect to the instance using Remote Desktop and verify that it boots into DSRM mode.

Note
Delete or stop the *temporary* instance you created in this procedure.

High CPU shortly after Windows starts

If Windows Update is set to **Check for updates but let me choose whether to download and install them** (the default instance setting) this check can consume anywhere from 50 - 99% of the CPU on the instance. If this CPU consumption causes problems for your applications, you can manually change Windows Update settings in **Control Panel** or you can use the following script in the Amazon EC2 user data field:

```
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Up
date" /v
          AUOptions /t REG_DWORD /d 3 /f net stop wuauserv net start wuauserv
```

When you execute this script specify a value for /d. The default value is 3. Possible values include the following:

1. Never check for updates
2. Check for updates but let me choose whether to download and install them
3. Download updates but let me choose whether to install them
4. Install updates automatically

To modify the user data for a Amazon EBS-backed instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Instances**, and select the instance.
3. Click **Actions**, select **Instance State**, and then click **Stop**.
4. In the confirmation dialog box, click **Yes, Stop**. It can take a few minutes for the instance to stop.
5. With the instance still selected, click **Actions**, select **Instance Settings**, and then click **View/Change User Data**. Note that you can't change the user data if the instance is running, but you can view it.
6. In the **View/Change User Data** dialog box, update the user data, and then click **Save**.

After you modify the user data for your instance, you can execute it. For more information, see Executing User Data (p. 243).

No console output

For Windows instances, the instance console displays the output from the EC2Config service running on the instance. The output logs the status of tasks performed during the Windows boot process. If Windows

boots successfully, the last message logged is `Windows is Ready to use`. Note that you can also display event log messages in the console, but this feature is not enabled by default. For more information, see Ec2 Service Properties (p. 237).

To get the console output for your instance using the Amazon EC2 console, select the instance, click **Actions**, select **Instance Settings**, and then click **Get System Log**. To get the console output using the command line, use one of the following commands: get-console-output (AWS CLI) or ec2-get-console-output (Amazon EC2 CLI).

If the console output is empty, it could indicate an issue with the EC2Config service, such as a misconfigured configuration file, or that Windows failed to boot properly. To fix the issue, download and install the latest version of EC2Config. For more information, see Installing the Latest Version of EC2Config (p. 256).

Instance terminates immediately

After you launch an instance, we recommend that you check its status to confirm that it goes from the `pending` status to the `running` status, and not the `terminated` status.

If the instance terminates immediately, you can use the Amazon EC2 console or command line to get information about the reason that the instance terminated.

To get the reason that an instance terminated using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances** to display the instance details.
3. Select your instance.
4. In the **Description** tab, locate the reason next to the label **State transition reason**. If the instance is still running, there's typically no reason listed. If you've explicitly stopped or terminated the instance, the reason is `User initiated shutdown`.

To get the reason that an instance terminated using the command line

Use the describe-instances command (AWS CLI) with the ID of the instance or the ec2-describe-instances command (Amazon EC2 CLI) with the ID of the instance and the `--verbose` option. Look for the `StateReason` element in the output.

"Password is not available"

To connect to a Windows instance using Remote Desktop, you must specify an account and password. The accounts and passwords provided are based on the AMI that you used to launch the instance. You can either retrieve the auto-generated password for the Administrator account, or use the account and password that were in use in the original instance from which the AMI was created.

If your Windows instance isn't configured to generate a random password, you'll receive the following message when you retrieve the auto-generated password using the console:

```
Password is not available.
The instance was launched from a custom AMI, or the default password has changed.
 A
password cannot be retrieved for this instance. If you have forgotten your
password, you can
```

```
reset it using the Amazon EC2 configuration service. For more information, see
 Passwords for a
Windows Server instance.
```

Check the console output for the instance to see whether the AMI that you used to launch it was created with password generation disabled. If password generation is disabled, the console output contains the following:

```
Ec2SetPassword: Disabled
```

If password generation is disabled and you don't remember the password for the original instance, you can reset the password for this instance. For more information, see Resetting an Administrator Password that's Lost or Expired (p. 278).

"Password not available yet"

To connect to a Windows instance using Remote Desktop, you must specify an account and password. The accounts and passwords provided are based on the AMI that you used to launch the instance. You can either retrieve the auto-generated password for the Administrator account, or use the account and password that were in use in the original instance from which the AMI was created.

Your password should be available within a few minutes. If the password isn't available, you'll receive the following message when you retrieve the auto-generated password using the console:

```
Password not available yet.
Please wait at least 4 minutes after launching an instance before trying to
retrieve the
auto-generated password.
```

If it's been longer than four minutes and you still can't get the password, it's possible that EC2Config is disabled. Verify by checking whether the console output is empty. For more information, see No console output (p. 690).

Also verify that the AWS Identity and Access Management (IAM) account being used to access the Management Portal has the ec2:GetPasswordData action allowed. For more information about IAM permissions, see What is IAM?.

"Cannot retrieve Windows password"

To retrieve the auto-generated password for the Administrator account, you must use the private key for the key pair that you specified when you launched the instance. If you didn't specify a key pair when you launched the instance, you'll receive the following message.

```
Cannot retrieve Windows password
```

You can terminate this instance and launch a new instance using the same AMI, making sure to specify a key pair.

"Waiting for the metadata service"

A Windows instance must obtain information from its instance metadata before it can activate itself. By default, the `WaitForMetaDataAvailable` setting ensures that the EC2Config service waits for the instance metadata to be accessible before continuing with the boot process. For more information, see Instance Metadata and User Data (p. 160).

If the instance is failing the instance reachability test, try the following to resolve this issue.

- [EC2-VPC] Check the CIDR block for your VPC. A Windows instance cannot boot correctly if it's launched into a VPC that has an IP address range from `224.0.0.0` to `255.255.255.255` (Class D and Class E IP address ranges). These IP address ranges are reserved, and should not be assigned to host devices. We recommend that you create a VPC with a CIDR block from the private (non-publicly routable) IP address ranges as specified in RFC 1918.
- It's possible that the system has been configured with a static IP address. Try the following:
  - [EC2-VPC] Create a network interface (p. 496) and attach it to the instance (p. 499).
  - [EC2-Classic] Enable DHCP.
- **To enable DHCP on a Windows instance that you can't connect to**

  1. Stop the affected instance and detach its root volume.
  2. Launch a temporary instance in the same Availability Zone as the affected instance.

     Warning
     If your temporary instance is based on the same AMI that the original instance is based on, and the operating system is later than Windows Server 2003, you must complete additional steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision. Alternatively, select a different AMI for the temporary instance. For example, if the original instance uses the AWS Windows AMI for Windows Server 2008 R2, launch the temporary instance using the AWS Windows AMI for Windows Server 2012 or Windows Server 2003. (To find an AMI for Windows Server 2003, search for an AMI using the name `Windows_Server-2003-R2_SP2`.)
  3. Attach the root volume from the affected instance to this temporary instance. Connect to the temporary instance, open the **Disk Management** utility, and bring the drive online.
  4. From the temporary instance, open **Regedit** and select **HKEY_LOCAL_MACHINE**. From the **File** menu, click **Load Hive**. Select the drive, open the file `Windows\System32\config\SYSTEM`, and specify a key name when prompted (you can use any name).
  5. Select the key that you just loaded and navigate to `ControlSet001\Services\Tcpip\Parameters\Interfaces`. Each network interface is listed by a GUID. Select the correct network interface. If DHCP is disabled and a static IP address assigned, `EnableDHCP` is set to 0. To enable DHCP, set `EnableDHCP` to 1, and delete the following keys if they exist: `NameServer`, `SubnetMask`, `IPAddress`, and `DefaultGateway`. Select the key again, and from the **File** menu, click **Unload Hive**.
  6. (Optional) If DHCP is already enabled, it's possible that you don't have a route to the metadata service. Updating EC2Config can resolve this issue.

     a. Download the latest Amazon Windows EC2Config Service. Extract the files from the `.zip` file to the `Temp` directory on the drive you attached.
     b. Open **Regedit** and select **HKEY_LOCAL_MACHINE**. From the **File** menu, click **Load Hive**. Select the drive, open the file `Windows\System32\config\SOFTWARE`, and specify a key name when prompted (you can use any name).
     c. Select the key that you just loaded and navigate to `Microsoft\Windows\CurrentVersion`. Select the `RunOnce` key. (If this key doesn't exist, right-click `CurrentVersion`, point to **New**, select **Key**, and name the key `RunOnce`.) Right-click, point to **New**, and select **String Value**. Enter `Ec2Install` as the name and `C:\Temp\Ec2Install.exe -q` as the data.

d. Select the key again, and from the **File** menu, click **Unload Hive**.

7. (Optional) If your temporary instance is based on the same AMI that the original instance is based on, and the operating system is later than Windows Server 2003, you must complete the following steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision.

   a. In the Registry Editor, load the following registry hive into a folder named `BCD`: `d:\boot\bcd`.
   b. Search for the following data value in BCD: "Windows Boot Manager". You'll find a match under a key named `12000004`.
   c. Select the key named `11000001` that is sibling to the key you found in the previous step. View the data for the `Element` value.
   d. Locate the four-byte disk signature at offset 0x38 in the data. Reverse the bytes to create the disk signature, and write it down. For example, the disk signature represented by the following data is `E9EB3AA5`:

```
...
0030  00 00 00 00 01 00 00 00
0038  A5 3A EB E9 00 00 00 00
0040  00 00 00 00 00 00 00 00
...
```

   e. In a Command Prompt window, run the following command to start Microsoft DiskPart.

```
C:\> diskpart
```

   f. Run the following DiskPart command to select the volume. (You can verify that the disk number is 1 using the **Disk Management** utility.)

```
DISKPART> select disk 1

Disk 1 is now the selected disk.
```

   g. Run the following DiskPart command to get the disk signature.

```
DISKPART> uniqueid disk

Disk ID: 0C764FA8
```

   h. If the disk signature shown in the previous step doesn't match the disk signature from BCD that you wrote down earlier, use the following DiskPart command to change the disk signature so that it matches:

```
DISKPART> uniqueid disk id=E9EB3AA5
```

8. Using the **Disk Management** utility, bring the drive offline.

   Note
   The drive is automatically offline if the temporary instance is running the same operating system as the affected instance, so you won't need to bring it offline manually.

9. Detach the volume from the temporary instance. You can terminate the temporary instance if you have no further use for it.
10. Restore the root volume of the affected instance by attaching the volume as `/dev/sda1`.
11. Start the affected instance.

If you are connected to the instance, open an Internet browser from the instance and enter the following URL for the metadata server:

```
http://169.254.169.254/latest/meta-data/
```

If you can't contact the metadata server, try the following to resolve the issue:

- Download and install the latest version of EC2Config. For more information, see Installing the Latest Version of EC2Config (p. 256).
- Check whether the Windows instance is running RedHat PV drivers. If so, update to Citrix PV drivers. For more information, see Upgrading PV Drivers on Your Windows AMI (p. 265).
- Verify that the firewall, IPSec, and proxy settings do not block outgoing traffic to the metadata service (`169.254.169.254`) or the KMS servers (the addresses are specified in `TargetKMSServer` elements in `C:\Program Files\Amazon\Ec2ConfigService\Settings\ActivationSettings.xml`).
- Verify that you have a route to the metadata service (`169.254.169.254`) using the following command.

```
C:\> route print
```

- Check for network issues that might affect the Availability Zone for your instance. Go to http://status.aws.amazon.com/.

Remote Desktop can't connect to the remote computer

Try the following to resolve issues related to connecting to your instance:

- Verify that you're using the correct public DNS hostname. (In the Amazon EC2 console, select the instance and check **Public DNS** in the details pane.) If your instance is in a VPC and you do not see a public DNS name, you must enable DNS hostnames. For more information, see Using DNS with Your VPC in the *Amazon VPC User Guide*.
- Verify that your security group has a rule that allows RDP access. For more information, see Create a Security Group (p. 17).
- If you copied the password but get the error "Your credentials did not work", try typing them manually when prompted. It's possible that you missed a character or got an extra whitespace character when you copied the password.
- Verify that the instance has passed status checks. For more information, see Status Checks for Your Instances (p. 322) and Troubleshooting Instances with Failed Status Checks (*Amazon EC2 User Guide for Linux Instances*).
- [EC2-VPC] Verify that the route table for the subnet has a route that sends all traffic destined outside the VPC (`0.0.0.0/0`) to the Internet gateway for the VPC. For more information, see Creating a Custom Route Table (Internet Gateways) in the *Amazon VPC User Guide*.

- Verify that Windows Firewall, or other firewall software, is not blocking RDP traffic to the instance. We recommend that you disable Windows Firewall and control access to your instance using security group rules.

To disable Windows Firewall on a Windows instance that you can't connect to

1. Stop the affected instance and detach its root volume.
2. Launch a temporary instance in the same Availability Zone as the affected instance.

   Warning
   If your temporary instance is based on the same AMI that the original instance is based on, and the operating system is later than Windows Server 2003, you must complete additional steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision. Alternatively, select a different AMI for the temporary instance. For example, if the original instance uses the AWS Windows AMI for Windows Server 2008 R2, launch the temporary instance using the AWS Windows AMI for Windows Server 2012 or Windows Server 2003. (To find an AMI for Windows Server 2003, search for an AMI using the name `Windows_Server-2003-R2_SP2`.)
3. Attach the root volume from the affected instance to this temporary instance. Connect to the temporary instance, open the **Disk Management** utility, and bring the drive online.
4. Open **Regedit** and select **HKEY_LOCAL_MACHINE**. From the **File** menu, click **Load Hive**. Select the drive, open the file `Windows\System32\config\SYSTEM`, and specify a key name when prompted (you can use any name).
5. Select the key you just loaded and navigate to `ControlSet001\Services\SharedAccess\Parameters\FirewallPolicy`. For each key with a name of the form *xxxx*Profile, select the key and change `EnableFirewall` from 1 to 0. Select the key again, and from the **File** menu, click **Unload Hive**.
6. (Optional) If your temporary instance is based on the same AMI that the original instance is based on, and the operating system is later than Windows Server 2003, you must complete the following steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision.

   a. In the Registry Editor, load the following registry hive into a folder named `BCD`: `d:\boot\bcd`.
   b. Search for the following data value in BCD: "Windows Boot Manager". You'll find a match under a key named `12000004`.
   c. Select the key named `11000001` that is sibling to the key you found in the previous step. View the data for the `Element` value.
   d. Locate the four-byte disk signature at offset 0x38 in the data. Reverse the bytes to create the disk signature, and write it down. For example, the disk signature represented by the following data is `E9EB3AA5`:

```
...
0030  00 00 00 00 01 00 00 00
0038  A5 3A EB E9 00 00 00 00
0040  00 00 00 00 00 00 00 00
...
```

   e. In a Command Prompt window, run the following command to start Microsoft DiskPart.

```
C:\> diskpart
```

   f. Run the following DiskPart command to select the volume. (You can verify that the disk number is 1 using the **Disk Management** utility.)

```
DISKPART> select disk 1

Disk 1 is now the selected disk.
```

g. Run the following DiskPart command to get the disk signature.

```
DISKPART> uniqueid disk

Disk ID: 0C764FA8
```

h. If the disk signature shown in the previous step doesn't match the disk signature from BCD that you wrote down earlier, use the following DiskPart command to change the disk signature so that it matches:

```
DISKPART> uniqueid disk id=E9EB3AA5
```

7. Using the **Disk Management** utility, bring the drive offline.

   Note
   The drive is automatically offline if the temporary instance is running the same operating system as the affected instance, so you won't need to bring it offline manually.

8. Detach the volume from the temporary instance. You can terminate the temporary instance if you have no further use for it.
9. Restore the root volume of the affected instance by attaching it as `/dev/sda1`.
10. Start the instance.

- Verify that the password has not expired. If the password has expired, you can reset it. For more information, see Resetting an Administrator Password that's Lost or Expired (p. 278).
- If you attempt to connect using a user account that you created on the instance and receive the error `The user cannot connect to the server due to insufficient access privileges`, verify that you granted the user the right to log on locally. For more information, see http://technet.microsoft.com/en-us/library/ee957044.aspx.
- If you attempt more than the maximum allowed concurrent RDP sessions, your session is terminated with the message `Your Remote Desktop Services session has ended. Another user connected to the remote computer, so your connection was lost.` By default, you are allowed two concurrent RDP sessions to your instance.

RDP displays a black screen instead of the desktop

Try the following to resolve this issue:

- Check the console output for additional information. To get the console output for your instance using the Amazon EC2 console, select the instance, click **Actions**, select **Instance Settings**, and then click **Get System Log**.
- Verify that you are running the latest version of your RDP client.

- Try the default settings for the RDP client. For more information, see Remote Session Environment in the *Microsoft TechNet Library*.
- If you are using Remote Desktop Connection, try starting it with the `/admin` option as follows.

```
C:\> mstsc /v:instance /admin
```

- If the server is running a full-screen application, it might have stopped responding. Use Ctrl+Shift+Esc to start Windows Task Manager, and then close the application.
- If the server is over-utilized, it might have stopped responding. To monitor the instance using the Amazon EC2 console, select the instance and then select the **Monitoring** tab. If you need to change the instance type to a larger size, see Resizing Your Instance (p. 118).

"Unable to activate Windows"

Windows instances use KMS for activation. You can receive this message, or `A problem occurred when Windows tried to activate. Error Code 0xC004F074`, if your instance can't reach the KMS server. Windows must be activated every 180 days. EC2Config attempts to contact the KMS server before the activation period expires to ensure that Windows remains activated.

Try the following to resolve issues activating Windows:

- Download and install the latest version of EC2Config. For more information, see Installing the Latest Version of EC2Config (p. 256).
- Verify that you are using the Amazon DNS server in addition to any other DNS servers you're using, or that the Amazon DNS server (`172.16.0.23`) is listed as a DNS forwarder.
- Verify that you have routes to the KMS servers. Open `C:\Program Files\Amazon\Ec2ConfigService\Settings\ActivationSettings.xml` and locate the `TargetKMSServer` elements. Run the following command and check whether the addresses for these KMS servers are listed.

```
C:\> route print
```

- Verify that the KMS client key is set. Run the following command and check the output.

```
C:\> C:\Windows\System32\slmgr.vbs /dlv
```

  If the output contains `Error: product key not found`, the KMS client key isn't set. If the KMS client key isn't set, look up the client key as described in this Microsoft TechNet article: http://technet.microsoft.com/en-us/library/jj612867.aspx, and then run the following command to set the KMS client key.

```
C:\> C:\Windows\System32\slmgr.vbs /ipk client_key
```

- Verify that the system has the correct time and time zone. If you are using Windows Server 2008 or later and a time zone other than UTC, add the following registry key and set it to `1` to ensure that the time is correct: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal**.
- If Windows Firewall is enabled, temporarily disable it using the following command.

```
C:\> netsh advfirewall set allprofiles state off
```

"Windows is not genuine (0x80070005)"

Windows instances use KMS for activation. If an instance is unable to complete the activation process, it reports that the copy of Windows is not genuine.

Try the suggestions for "Unable to activate Windows" (p. 698).

"No Terminal Server License Servers available to provide a license"

By default, Windows Server is licensed for two simultaneous users through Remote Desktop. If you need to provide more than two users with simultaneous access to your Windows instance through Remote Desktop, you can purchase a Remote Desktop Services client access license (CAL) and install the Remote Desktop Session Host and Remote Desktop Licensing Server roles.

Check for the following issues:

- You've exceeded the maximum number of concurrent RDP sessions.
- You've installed the Windows Remote Desktop Services role.
- Licensing has expired. If the licensing has expired, you can't connect to your Windows instance as a user. You can try the following:
  - Connect to the instance from the command line using an `/admin` parameter, for example:

    ```
    C:\> mstsc /v:instance /admin
    ```

    For more information, go to the following Microsoft article: Use command line parameters with Remote Desktop Connection.
  - Stop the instance, detach its Amazon EBS volumes, and attach them to another instance in the same Availability Zone to recover your data.

Instance loses network connectivity or scheduled tasks don't run when expected

If you restart your instance and it loses network connectivity, it's possible that the instance has the wrong time.

By default, Windows instances use Coordinated Universal Time (UTC). If you set the time for your instance to a different time zone and then restart it, the time becomes offset and the instance temporarily loses its IP address. The instance regains network connectivity eventually, but this can take several hours. The amount of time that it takes for the instance to regain network connectivity depends on the difference between UTC and the other time zone.

This same time issue can also result in scheduled tasks not running when you expect them to. In this case, the scheduled tasks do not run when expected because the instance has the incorrect time.

To use a time zone other than UTC persistently, you must set the **RealTimeIsUniversal** registry key. Without this key, an instance uses UTC after you restart it.

Important
Windows Server 2003 doesn't support the **RealTimeIsUniversal** registry key. Therefore, the instance always uses UTC after a restart.

To resolve time issues that cause a loss of network connectivity

1. Ensure that you are running the recommended PV drivers. For more information, see Upgrading PV Drivers on Your Windows AMI (p. 265).
2. Verify that the following registry key exists and is set to 1:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal

Insufficient Instance Capacity

If you get an `InsufficientInstanceCapacity` error when you try to launch an instance, AWS does not currently have enough available capacity to service your request.

Try the following:

- Wait a few minutes and then submit your request again; capacity can shift frequently.
- Submit a new request with a reduced number of instances. For example, if you're making a single request to launch 15 instances, try making 3 requests for 5 instances, or 15 requests for 1 instance instead.
- Submit a new request without specifying an Availability Zone.
- Submit a new request using a different instance type (which you can resize at a later stage). For more information, see Resizing Your Instance (p. 118).
- Try purchasing Reserved Instances. Reserved Instances are a long-term capacity reservation. For more information, see Amazon EC2 Reserved Instances.

Instance Limit Exceeded

If you get an `InstanceLimitExceeded` error when you try to launch an instance, you have reached your concurrent running instance limit. For new AWS accounts, the default limit is 20. If you need additional running instances, complete the form at Request to Increase Amazon EC2 Instance Limit.

Windows Server 2012 R2 not available on the network

For information about troubleshooting a Windows Server 2012 R2 instance that is not available on the network, see Windows Server 2012 R2 loses network and storage connectivity after an instance reboot (p. 270).

Document History

The following table describes important additions to the Amazon EC2 documentation. We also update the documentation frequently to address the feedback that you send us.

Current API version: 2015-04-15.

Feature	API Version	Description	Release Date
Spot fleet diversified allocation strategy		You can now allocate Spot instances in multiple Spot pools using a single Spot fleet request. For more information, see Spot Fleet Allocation Strategy (p. 127).	15 September 2015
Spot fleet instance weighting		You can now define the capacity units that each instance type contributes to your application's performance, and adjust your bid price for each Spot pool accordingly. For more information, see Spot Fleet Instance Weighting (p. 128).	31 August 2015
New reboot alarm action and new IAM role for use with alarm actions		Added the reboot alarm action and new IAM role for use with alarm actions. For more information, see Create Alarms That Stop, Terminate, Reboot, or Recover an Instance (p. 368).	23 July 2015
New `t2.large` instance type		T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see T2 Instances (p. 99).	16 June 2015
M4 instances		The next generation of general-purpose instances that provide a balance of compute, memory, and network resources. M4 instances are powered by a custom Intel 2.4 GHz Intel® Xeon® E5 2676v3 (Haswell) processor with AVX2.	11 June 2015

Feature	API Version	Description	Release Date
Spot fleets	2015-04-15	You can manage a collection, or fleet, of Spot instances instead of managing separate Spot instance requests. For more information, see How Spot Fleet Works (p. 127).	18 May 2015
Migrate Elastic IP addresses to EC2-Classic	2015-04-15	You can migrate an Elastic IP address that you've allocated for use in the EC2-Classic platform to the EC2-VPC platform. For more information, see Migrating an Elastic IP Address from EC2-Classic to EC2-VPC (p. 487).	15 May 2015
Importing VMs with multiple disks as AMIs	2015-03-01	The VM Import process now supports the importation VMs with multiple disks as AMIs. For more information, see Importing and Exporting Instances (p. 169).	23 April 2015
New `g2.8xlarge` instance type		The new `g2.8xlarge` instance is backed by four high-performance NVIDIA GPUs, making it well suited for GPU compute workloads including large scale rendering, transcoding, machine learning, and other server-side workloads that require massive parallel processing power.	7 April 2015
D2 instances		Next generation Amazon EC2 dense-storage instances that are optimized for applications requiring sequential access to large amount of data on direct attached instance storage. D2 instances are designed to offer best price/performance in the dense-storage family. Powered by 2.4 GHz Intel® Xeon® E5 2676v3 (Haswell) processors, D2 instances improve on HS1 instances by providing additional compute power, more memory, and Enhanced Networking. In addition, D2 instances are available in four instance sizes with 6TB, 12TB, 24TB, and 48TB storage options. For more information, see D2 Instances (p. 107).	24 March 2015
Amazon EC2 Simple Systems Manager (SSM)		SSM enables you to configure and manage your EC2 instances. For more information, see Managing Windows Instance Configuration (p. 284) and Joining a Windows Instance to an AWS Directory Service Domain (p. 291).	17 February 2015
AWS Systems Manager for Microsoft SCVMM 1.5		You can now use AWS Systems Manager for Microsoft SCVMM to launch an instance and to import a VM from SCVMM to Amazon EC2. For more information, see Creating an EC2 Instance (p. 635) and Importing Your Virtual Machine (p. 640).	21 January 2015

Feature	API Version	Description	Release Date
Automatic recovery for EC2 instances		You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair. A recovered instance is identical to the original instance, including the instance ID, IP addresses, and all instance metadata. For more information, see Recover Your Instance (p. 229).	12 January 2015
C4 instances		Next-generation compute-optimized instances that provide very high CPU performance at an economical price. C4 instances are based on custom 2.9 GHz Intel® Xeon® E5-2666 v3 (Haswell) processors. With additional Turbo boost, the processor clock speed in C4 instances can reach as high as 3.5Ghz with 1 or 2 core turbo. Expanding on the capabilities of C3 compute-optimized instances, C4 instances offer customers the highest processor performance among EC2 instances. These instances are ideally suited for high-traffic web applications, ad serving, batch processing, video encoding, distributed analytics, high-energy physics, genome analysis, and computational fluid dynamics. For more information, see C4 Instances (p. 102).	11 January 2015
ClassicLink	2014-10-01	ClassicLink enables you to link your EC2-Classic instance to a VPC in your account. You can associate VPC security groups with the EC2-Classic instance, enabling communication between your EC2-Classic instance and instances in your VPC using private IP addresses. For more information, see ClassicLink (p. 457).	7 January 2015
Spot instance termination notices		The best way to protect against Spot instance interruption is to architect your application to be fault tolerant. In addition, you can take advantage of Spot instance termination notices, which provide a two-minute warning before Amazon EC2 must terminate your Spot instance. For more information, see Spot Instance Termination Notices (p. 156).	5 January 2015
AWS Systems Manager for Microsoft SCVMM		AWS Systems Manager for Microsoft SCVMM provides a simple, easy-to-use interface for managing AWS resources, such as EC2 instances, from Microsoft SCVMM. For more information, see AWS Systems Manager for Microsoft System Center VMM (p. 630).	29 October 2014

Feature	API Version	Description	Release Date
`DescribeVolumes` pagination support	2014-09-01	The `DescribeVolumes` API call now supports the pagination of results with the `MaxResults` and `NextToken` parameters. For more information, see DescribeVolumes in the *Amazon EC2 API Reference*.	23 October 2014
Added support for Amazon CloudWatch Logs		You can use Amazon CloudWatch Logs to monitor, store, and access your system, application, and custom log files from your instances or other sources. You can then retrieve the associated log data from CloudWatch Logs using the Amazon CloudWatch console, the CloudWatch Logs commands in the AWS CLI, or the CloudWatch Logs SDK. For more information, see Configuring a Windows Instance Using the EC2Config Service (p. 235). For more information about CloudWatch Logs, see Monitoring System, Application, and Custom Log Files in the Amazon CloudWatch Developer Guide.	10 July 2014
T2 instances	2014-06-15	T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see T2 Instances (p. 99).	30 June 2014
New **EC2 Service Limits** page		Use the **EC2 Service Limits** page in the Amazon EC2 console to view the current limits for resources provided by Amazon EC2 and Amazon VPC, on a per-region basis.	19 June 2014
Amazon EBS General Purpose (SSD) Volumes	2014-05-01	General Purpose (SSD) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies, the ability to burst to 3,000 IOPS for extended periods of time, and a base performance of 3 IOPS/GiB. General Purpose (SSD) volumes can range in size from 1 GiB to 1 TiB. For more information, see General Purpose (SSD) Volumes (p. 521).	16 June 2014
Windows Server 2012 R2		AMIs for Windows Server 2012 R2 use the new AWS PV drivers. For more information, see AWS PV Drivers (p. 262).	3 June 2014
AWS Management Pack		AWS Management Pack now supports for System Center Operations Manager 2012 R2. For more information, see AWS Management Pack for Microsoft System Center (p. 644).	22 May 2014

Feature	API Version	Description	Release Date
Amazon EBS encryption	2014-05-01	Amazon EBS encryption offers seamless encryption of EBS data volumes and snapshots, eliminating the need to build and maintain a secure key management infrastructure. EBS encryption enables data at rest security by encrypting your data using Amazon-managed keys. The encryption occurs on the servers that host EC2 instances, providing encryption of data as it moves between EC2 instances and EBS storage. For more information, see Amazon EBS Encryption (p. 558).	21 May 2014
R3 instances	2014-02-01	Next generation memory-optimized instances with the best price point per GiB of RAM and high performance. These instances are ideally suited for relational and NoSQL databases, in-memory analytics solutions, scientific computing, and other memory-intensive applications that can benefit from the high memory per vCPU, high compute performance, and enhanced networking capabilities of R3 instances. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances.	9 April 2014
Amazon EC2 Usage Reports		Amazon EC2 Usage Reports is a set of reports that shows cost and usage data of your usage of EC2. For more information, see Amazon EC2 Usage Reports (p. 619).	28 January 2014
Additional M3 instances	2013-10-15	The M3 instance sizes `m3.medium` and `m3.large` are now supported. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances.	20 January 2014
I2 instances	2013-10-15	These instances provide very high IOPS. I2 instances also support enhanced networking that delivers improve inter-instance latencies, lower network jitter, and significantly higher packet per second (PPS) performance. For more information, see I2 Instances (p. 106).	19 December 2013
Updated M3 instances	2013-10-15	The M3 instance sizes, `m3.xlarge` and `m3.2xlarge` now support instance store with SSD volumes. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances.	19 December 2013
Resource-level permissions for RunInstances	2013-10-15	You can now create policies in AWS Identity and Access Management to control resource-level permissions for the Amazon EC2 RunInstances API action. For more information and example policies, see Controlling Access to Amazon EC2 Resources (p. 406).	20 November 2013

Feature	API Version	Description	Release Date
C3 instances	2013-10-15	Compute-optimized instances that provide very high CPU performance at an economical price. C3 instances also support enhanced networking that delivers improved inter-instance latencies, lower network jitter, and significantly higher packet per second (PPS) performance. These instances are ideally suited for high-traffic web applications, ad serving, batch processing, video encoding, distributed analytics, high-energy physics, genome analysis, and computational fluid dynamics. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances.	14 November 2013
Launching an instance from the AWS Marketplace		You can now launch an instance from the AWS Marketplace using the Amazon EC2 launch wizard. For more information, see Launching an AWS Marketplace Instance (p. 213).	11 November 2013
G2 instances	2013-10-01	These instances are ideally suited for video creation services, 3D visualizations, streaming graphics-intensive applications, and other server-side workloads requiring massive parallel processing power. For more information, see Windows GPU Instances (p. 104).	4 November 2013
New launch wizard		There is a new and redesigned EC2 launch wizard. For more information, see Launching an Instance (p. 207).	10 October 2013
Modifying Amazon EC2 Reserved Instances	2013-08-15	You can now modify Reserved Instances in a region.	11 September 2013
Assigning a public IP address	2013-07-15	You can now assign a public IP address when you launch an instance in a VPC. For more information, see Assigning a Public IP Address (p. 479).	20 August 2013
Granting resource-level permissions	2013-06-15	Amazon EC2 supports new Amazon Resource Names (ARNs) and condition keys. For more information, see IAM Policies for Amazon EC2 (p. 408).	8 July 2013
Incremental Snapshot Copies	2013-02-01	You can now perform incremental snapshot copies. For more information, see Copying an Amazon EBS Snapshot (p. 552).	11 June 2013
AWS Management Pack		The AWS Management Pack links Amazon EC2 instances and the Microsoft Windows or Linux operating systems running inside them. The AWS Management Pack is an extension to Microsoft System Center Operations Manager. For more information, see AWS Management Pack for Microsoft System Center (p. 644).	8 May 2013

Feature	API Version	Description	Release Date
New **Tags** page		There is a new **Tags** page in the Amazon EC2 console. For more information, see Tagging Your Amazon EC2 Resources (p. 609).	04 April 2013
Additional EBS-optimized instance types	2013-02-01	The following instance types can now be launched as EBS-optimized instances: `c1.xlarge`, `m2.2xlarge`, `m3.xlarge`, and `m3.2xlarge`. For more information, see Amazon EBS–Optimized Instances (p. 555).	19 March 2013
PV Drivers		To learn how to upgrade the paravirtualized (PV) drivers on your Windows AMI, see Upgrading PV Drivers on Your Windows AMI (p. 265).	March 2013
AWS Diagnostics for Microsoft Windows Server		The topic AWS Diagnostics for Microsoft Windows Server - Beta (p. 682) describes how to diagnose and troubleshoot possible issues using the AWS Diagnostics for Microsoft Windows Server.	March 2013
Copy an AMI from one region to another	2013-02-01	You can copy an AMI from one region to another, enabling you to launch consistent instances in more than one AWS region quickly and easily. For more information, see Copying an AMI (p. 74).	11 March 2013
Launch instances into a default VPC	2013-02-01	Your AWS account is capable of launching instances into either the EC2-Classic or EC2-VPC platform, or only into the EC2-VPC platform, on a region-by-region basis. If you can launch instances only into EC2-VPC, we create a default VPC for you. When you launch an instance, we launch it into your default VPC, unless you create a nondefault VPC and specify it when you launch the instance. For more information, see Supported Platforms (p. 455).	11 March 2013
High-memory cluster (cr1.8xlarge) instance type	2012-12-01	Have large amounts of memory coupled with high CPU and network performance. These instances are well suited for in-memory analytics, graph analysis, and scientific computing applications.	21 January 2013
High storage (`hs1.8xlarge`) instance type	2012-12-01	High storage instances provide very high storage density and high sequential read and write performance per instance. They are well-suited for data warehousing, Hadoop/MapReduce, and parallel file systems. For more information, see HS1 Instances (p. 110).	20 December 2012
EBS snapshot copy	2012-12-01	You can use snapshot copies to create backups of data, to create new Amazon EBS volumes, or to create Amazon Machine Images (AMIs). For more information, see Copying an Amazon EBS Snapshot (p. 552).	17 December 2012

Feature	API Version	Description	Release Date
Updated EBS metrics and status checks for Provisioned IOPS (SSD) volumes	2012-10-01	Updated the EBS metrics to include two new metrics for Provisioned IOPS (SSD) volumes. For more information, see Monitoring Volumes with CloudWatch (p. 532). Also added new status checks for Provisioned IOPS (SSD) volumes. For more information, see Monitoring Volumes with Status Checks (p. 535).	20 November 2012
Support for Microsoft Windows Server 2012		Amazon EC2 now provides you with several preconfigured Windows Server 2012 AMIs. These AMIs are immediately available for use in every region and for every 64-bit instance type. The AMIs support the following languages: • English • Chinese Simplified • Chinese Traditional • Chinese Traditional Hong Kong • Japanese • Korean • Portuguese • Portuguese Brazil • Czech • Dutch • French • German • Hungarian • Italian • Polish • Russian • Spanish • Swedish • Turkish	19 November 2012
M3 instances	2012-10-01	There are new M3 extra-large and M3 double-extra-large instance types. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances.	31 October 2012
Spot instance request status	2012-10-01	Spot instance request status makes it easy to determine the state of your Spot requests.	14 October 2012

Feature	API Version	Description	Release Date
Amazon EC2 Reserved Instance Marketplace	2012-08-15	The Reserved Instance Marketplace matches sellers who have Amazon EC2 Reserved Instances that they no longer need with buyers who are looking to purchase additional capacity. Reserved Instances bought and sold through the Reserved Instance Marketplace work like any other Reserved Instances, except that they can have less than a full standard term remaining and can be sold at different prices.	11 September 2012
Provisioned IOPS (input/output operations per second) (SSD) for Amazon EBS	2012-07-20	Provisioned IOPS (SSD) volumes deliver predictable, high performance for I/O intensive workloads, such as database applications, that rely on consistent and fast response times. For more information, see Amazon EBS Volume Types (p. 520).	31 July 2012
High I/O instances for Amazon EC2	2012-06-15	High I/O instances provides very high, low latency, disk I/O performance using SSD-based local instance storage. For more information, see HI1 Instances (p. 108).	18 July 2012
IAM roles on Amazon EC2 instances	2012-06-01	IAM roles for Amazon EC2 provide: • AWS access keys for applications running on Amazon EC2 instances. • Automatic rotation of the AWS access keys on the Amazon EC2 instance. • Granular permissions for applications running on Amazon EC2 instances that make requests to your AWS services.	11 June 2012
Spot instance features that make it easier to get started and handle the potential of interruption.		You can now manage your Spot instances as follows: • Place bids for Spot instances using Auto Scaling launch configurations, and set up a schedule for placing bids for Spot instances. For more information, see Launching Spot Instances in Your Auto Scaling Group in the *Auto Scaling Developer Guide*. • Get notifications when instances are launched or terminated. • Use AWS CloudFormation templates to launch Spot instances in a stack with AWS resources.	7 June 2012
EC2 instance export and timestamps for status checks for Amazon EC2	2012-05-01	Added support for exporting Windows Server instances that you originally imported into EC2. Added support for timestamps on instance status and system status to indicate the date and time that a status check failed.	25 May 2012

Feature	API Version	Description	Release Date
EC2 instance export, and timestamps in instance and system status checks for Amazon VPC	2012-05-01	Added support for EC2 instance export to Citrix Xen, Microsoft Hyper-V, and VMware vSphere. Added support for timestamps in instance and system status checks.	25 May 2012
Cluster Compute Eight Extra Large instances	2012-04-01	Added support for `cc2.8xlarge` instances in a VPC.	26 April 2012
AWS Marketplace AMIs	2012-04-01	Added support for AWS Marketplace AMIs.	19 April 2012
Medium instances, support for 64-bit on all AMIs	2011-12-15	Added support for a new instance type and 64-bit information.	7 March 2012
Reserved Instance pricing tiers	2011-12-15	Added a new section discussing how to take advantage of the discount pricing that is built into the Reserved Instance pricing tiers.	5 March 2012
Elastic Network Interfaces (ENIs) for EC2 instances in Amazon Virtual Private Cloud	2011-12-01	Added new section about elastic network interfaces (ENIs) for EC2 instances in a VPC. For more information, see Elastic Network Interfaces (ENI) (p. 492).	21 December 2011
New offering types for Amazon EC2 Reserved Instances	2011-11-01	You can choose from a variety of Reserved Instance offerings that address your projected use of the instance: *Heavy Utilization*, *Medium Utilization*, and *Light Utilization*.	01 December 2011
Amazon EC2 instance status	2011-11-01	You can view additional details about the status of your instances, including scheduled events planned by AWS that might have an impact on your instances. These operational activities include instance reboots required to apply software updates or security patches, or instance retirements required where there are hardware issues. For more information, see Monitoring the Status of Your Instances (p. 322).	16 November 2011
Amazon EC2 Cluster Compute Instance Type		Added support for Cluster Compute Eight Extra Large (cc2.8xlarge) to Amazon EC2.	14 November 2011
Spot instances in Amazon VPC	2011-07-15	Added information about the support for Spot instances in Amazon VPC. With this update, users can launch Spot instances a virtual private cloud (VPC). By launching Spot instances in a VPC, users of Spot instances can enjoy the benefits of Amazon VPC.	11 October 2011

Feature	API Version	Description	Release Date
Simplified VM import process for users of the CLI tools	2011-07-15	The VM Import process for CLI users is simplified with the enhanced functionality of `ec2-import-instance` and `ec2-import-volume`, which now will perform the upload of the images into Amazon EC2 after creating the import task. In addition, with the introduction of the `ec2-resume-import` command, users can restart an incomplete upload at the point the task stopped. For more information, see Step 4: Importing Your VM into Amazon EC2 (p. 189).	15 September 2011
Support for importing in VHD file format		VM Import can now import virtual machine image files in VHD format. The VHD file format is compatible with the Citrix Xen and Microsoft Hyper-V virtualization platforms. With this release, VM Import now supports RAW, VHD and VMDK (VMware ESX-compatible) image formats. For more information, see Step 1: Install the Amazon EC2 CLI (p. 186).	24 August 2011
Support for Microsoft Windows Server 2003 R2		VM Import now supports Windows Server 2003 (R2). With this release, VM Import supports all versions of Microsoft Windows Server supported by Amazon EC2.	24 August 2011
Update to the Amazon EC2 VM Import Connector for VMware vCenter		Added information about the 1.1 version of the Amazon EC2 VM Import Connector for VMware vCenter virtual appliance (Connector). This update includes proxy support for Internet access, better error handling, improved task progress bar accuracy, and several bug fixes. For more information, see Importing a VM into Amazon EC2 Using ImportInstance (p. 185).	27 June 2011
Spot instances Availability Zone pricing changes	2011-05-15	Added information about the Spot instances Availability Zone pricing feature. In this release, we've added new Availability Zone pricing options as part of the information returned when you query for Spot instance requests and Spot price history. These additions make it easier to determine the price required to launch a Spot instance into a particular Availability Zone.	26 May 2011
AWS Identity and Access Management		Added information about AWS Identity and Access Management (IAM), which enables users to specify which Amazon EC2 actions a user can use with Amazon EC2 resources in general. For more information, see Controlling Access to Amazon EC2 Resources (p. 406).	26 April 2011

Feature	API Version	Description	Release Date
Dedicated instances		Launched within your Amazon Virtual Private Cloud (Amazon VPC), Dedicated Instances are instances that are physically isolated at the host hardware level. Dedicated Instances let you take advantage of Amazon VPC and the AWS cloud, with benefits including on-demand elastic provisioning and pay only for what you use, while isolating your Amazon EC2 compute instances at the hardware level. For more information, see Using EC2 Dedicated Instances in the *Amazon VPC User Guide*.	27 March 2011
Reserved Instances updates to the AWS Management Console		Updates to the AWS Management Console make it easier for users to view their Reserved Instances and purchase additional Reserved Instances, including Dedicated Reserved Instances.	27 March 2011
Support for Windows Server 2008 R2		Amazon EC2 now provides you with several preconfigured Windows Server 2008 R2 AMIs. These AMIs are immediately available for use in every region and in most 64-bit instance types, excluding t1.micro and HPC families. The AMIs will support multiple languages.	15 March 2011
Metadata information	2011-01-01	Added information about metadata to reflect changes in the 2011-01-01 release. For more information, see Instance Metadata and User Data (p. 160) and Instance Metadata Categories (p. 164).	11 March 2011
Amazon EC2 VM Import Connector for VMware vCenter		Added information about the Amazon EC2 VM Import Connector for VMware vCenter virtual appliance (Connector). The Connector is a plug-in for VMware vCenter that integrates with VMware vSphere Client and provides a graphical user interface that you can use to import your VMware virtual machines to Amazon EC2. For more information, see Importing a VM into Amazon EC2 Using ImportInstance (p. 185).	3 March 2011
Force volume detachment		You can now use the AWS Management Console to force the detachment of an Amazon EBS volume from an instance. For more information, see Detaching an Amazon EBS Volume from an Instance (p. 542).	23 February 2011
Instance termination protection		You can now use the AWS Management Console to prevent an instance from being terminated. For more information, see Enabling Termination Protection for an Instance (p. 226).	23 February 2011
VM Import	2010-11-15	Added information about VM Import, which allows you to import a virtual machine or volume into Amazon EC2. For more information, see Step 1: Install the Amazon EC2 CLI (p. 186).	15 December 2010

Feature	API Version	Description	Release Date
Basic monitoring for instances	2010-08-31	Added information about basic monitoring for EC2 instances.	12 December 2010
Cluster GPU instances	2010-08-31	Amazon EC2 offers cluster GPU instances (cg1.4xlarge) for high-performance computing (HPC) applications. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances.	14 November 2010
Filters and Tags	2010-08-31	Added information about listing, filtering, and tagging resources. For more information, see Listing and Filtering Your Resources (p. 606) and Tagging Your Amazon EC2 Resources (p. 609).	19 September 2010
Idempotent Instance Launch	2010-08-31	Added information about ensuring idempotency when running instances.	19 September 2010
Micro instances	2010-06-15	Amazon EC2 offers the `t1.micro` instance type for certain types of applications. For more information, see T1 Micro Instances (p. 111).	8 September 2010
AWS Identity and Access Management for Amazon EC2		Amazon EC2 now integrates with AWS Identity and Access Management (IAM). For more information, see Controlling Access to Amazon EC2 Resources (p. 406).	2 September 2010
Cluster instances	2010-06-15	Amazon EC2 offers cluster compute instances for high-performance computing (HPC) applications. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances.	12 July 2010
Amazon VPC IP Address Designation	2010-06-15	Amazon VPC users can now specify the IP address to assign an instance launched in a VPC.	12 July 2010
Amazon CloudWatch Monitoring for Amazon EBS Volumes		Amazon CloudWatch monitoring is now automatically available for Amazon EBS volumes. For more information, see Monitoring Volumes with CloudWatch (p. 532).	14 June 2010
High-memory extra large instances	2009-11-30	Amazon EC2 now supports a High-Memory Extra Large (m2.xlarge) instance type. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances.	22 February 2010
Reserved Instances with Windows		Amazon EC2 now supports Reserved Instances with Windows.	22 February 2010

AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the *AWS General Reference*.