
Amazon Elastic Compute Cloud

User Guide for Microsoft Windows

API Version 2014-06-15



Amazon Elastic Compute Cloud: User Guide for Microsoft Windows

Copyright © 2014 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

The following are trademarks of Amazon Web Services, Inc.: Amazon, Amazon Web Services Design, AWS, Amazon CloudFront, Cloudfront, CloudTrail, Amazon DevPay, DynamoDB, ElastiCache, Amazon EC2, Amazon Elastic Compute Cloud, Amazon Glacier, Kinesis, Kindle, Kindle Fire, AWS Marketplace Design, Mechanical Turk, Amazon Redshift, Amazon Route 53, Amazon S3, Amazon VPC. In addition, Amazon.com graphics, logos, page headers, button icons, scripts, and service names are trademarks, or trade dress of Amazon in the U.S. and/or other countries. Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon.

All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

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
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	33
Tutorial: Set Up a Windows HPC Cluster	35
Prerequisites	35
Task 1: Set Up Your Active Directory Domain Controller	35
Creating Security Groups for Active Directory	36
Creating the Domain Controller for your HPC cluster	36
Configuring the Domain Controller for Your HPC Cluster	37
Task 2: Configure Your Head Node	37
Creating Security Groups for Your HPC Cluster	37
Launch an Instance for the HPC Head Node	38
Install the HPC Pack	38
Configure Your HPC Cluster on the Head Node	39
Task 3: Set Up the Compute Node	39
Launch an Instance for the HPC Compute Node	39
Install the HPC Pack on the Compute Node	40
Add the Compute Node to Your HPC Cluster	40
Task 4: Scale Your HPC Compute Nodes (Optional)	41
Running the Lizard Performance Measurement Application	42
Create_AD_security.bat	42
Create-HPC-sec-group.bat	43
Amazon Machine Images	45

Using an AMI	45
Creating Your Own AMI	46
Buying, Sharing, and Selling AMIs	46
Deregistering Your AMI	46
AWS Windows AMIs	46
Update Schedule	47
Configuration Settings	47
Xen Drivers	47
Keeping Your Instances Up-to-Date	47
AMI Types	48
Launch Permissions	48
Storage for the Root Device	48
Finding an AMI	51
Finding a Windows AMI Using the Amazon EC2 Console	51
Finding an AMI Using the Command Line	52
Shared AMIs	53
Finding Shared AMIs	53
Making an AMI Public	55
Sharing an AMI with Specific AWS Accounts	56
Using Bookmarks	58
Paid AMIs	58
Selling Your AMI	59
Finding a Paid AMI	59
Purchase a Paid AMI	60
Getting the Product Code for Your Instance	60
Using Paid Support	61
Bills for Paid and Supported AMIs	61
Managing Your AWS Marketplace Subscriptions	61
Creating an Amazon EBS-Backed Windows AMI	62
Creating an AMI from an Instance	62
Creating an Instance Store-Backed Windows AMI	64
Instance Store-Backed Windows AMIs	64
Preparing to Create an Instance Store-Backed Windows AMI	65
Bundling an Instance Store-Backed Windows Instance	66
Registering an Instance Store-Backed Windows AMI	67
Copying an AMI	67
AMI Copy	68
Copying an Amazon EC2 AMI	68
Copying an Amazon EC2 AMI with Encrypted Volumes	69
Stopping a Pending AMI Copy Operation	71
Deregistering Your AMI	72
Cleaning Up Your Amazon EBS-Backed AMI	72
Cleaning Up Your Instance Store-Backed AMI	73
Instances	74
Instance Types	74
Available Instance Types	75
Hardware Specifications	76
T2 Instances	76
I2 Instances	79
H1 Instances	81
HS1 Instances	82
R3 Instances	83
GPU Instances	85
T1 Micro Instances	86
EBS-Optimized Instances	93
Placement Groups	94
Resizing Instances	97
Instance Metadata and User Data	100

Retrieving Instance Metadata	100
Retrieving User Data	102
Retrieving Dynamic Data	103
Instance Metadata Categories	103
Importing and Exporting Instances	107
Prerequisites	108
Importing a VM into Amazon EC2	111
Exporting Amazon EC2 Instances	120
Troubleshooting	121
Instance Lifecycle	126
Instance Launch	126
Instance Stop and Start (Amazon EBS-backed instances only)	127
Instance Reboot	127
Instance Retirement	127
Instance Termination	128
Differences Between Reboot, Stop, and Terminate	128
Launch	129
Launching an Instance	130
Launching an Instance from a Backup	136
Launching an AWS Marketplace Instance	136
Connect	138
Prerequisites	138
Connecting to Windows	138
Transfer Files to Windows Server Instances from Windows	140
Stop and Start	140
Overview	140
Stopping and Starting Your Instances	141
Modifying a Stopped Instance	142
Reboot	143
Retire	143
Identifying Instances Scheduled for Retirement	144
Working with Instances Scheduled for Retirement	144
Terminate	145
Instance Termination	146
Terminating an Instance	146
Enabling Termination Protection	147
Changing the Shutdown Behavior	148
Preserving Amazon EBS Volumes on Instance Termination	148
Configure Instances	151
Using EC2Config	151
Overview of EC2Config Tasks	152
Ec2 Service Properties	153
EC2Config Settings Files	159
Installing the Latest Version of EC2Config	162
Stopping, Deleting, or Uninstalling EC2Config	163
Troubleshooting CloudWatch Logs in EC2Config	164
Upgrading PV Drivers	164
Xen Drivers	165
Upgrading PV Drivers on Your Windows Server 2008 and 2008 R2 Instances	167
Upgrading Your Citrix Xen Guest Agent Service	169
Upgrading PV Drivers on Your Windows Server 2003 Instance	170
Troubleshooting	171
Setting the Password	174
Changing the Administrator Password After Connecting	175
Resetting an Administrator Password that's Lost or Expired	175
Configuring a Secondary Private IP Address	179
Prerequisites	179
Step 1: Configure Static IP Addressing on Your Windows Instance	179

Step 2: Configure a Secondary Private IP Address for Your Windows Instance	181
Step 3: Configure Applications to Use the Secondary Private IP Address	182
Setting the Time	183
Changing the Time Zone	183
Configuring Network Time Protocol (NTP)	184
Configuring Time Settings for Windows Server 2008 and later	184
Configuring Time Settings for Windows Server 2003	185
Monitoring	186
Automated and Manual Monitoring	187
Automated Monitoring Tools	187
Manual Monitoring Tools	188
Best Practices for Monitoring	189
Monitoring the Status of Your Instances	189
Monitoring Instances with Status Checks	189
Monitoring Events for Your Instances	194
Monitoring Your Instances with CloudWatch	197
Enabling or Disabling Detailed Monitoring on an Amazon EC2 Instance	198
View Amazon EC2 Metrics	201
Get Statistics for Metrics	207
Graphing Metrics	223
Create a CloudWatch Alarm	227
Create Alarms That Stop or Terminate an Instance	234
Monitoring Scripts for Amazon EC2 Instances	249
Amazon CloudWatch Monitoring Scripts for Windows	249
Network and Security	259
Key Pairs	260
Creating Your Key Pair Using Amazon EC2	260
Importing Your Own Key Pair to Amazon EC2	261
Retrieving the Public Key for Your Key Pair	263
Verifying Your Key Pair's Fingerprint	263
Deleting Your Key Pair	264
Security Groups	264
Security Groups for EC2-Classic	265
Security Groups for EC2-VPC	265
Security Group Rules	265
Default Security Groups	266
Custom Security Groups	267
Creating a Security Group	268
Describing Your Security Groups	268
Adding Rules to a Security Group	269
Deleting Rules from a Security Group	270
Deleting a Security Group	270
API and Command Overview	270
Controlling Access	272
Network Access to Your Instance	272
Amazon EC2 Permission Attributes	272
IAM and Amazon EC2	272
IAM Policies	274
IAM Roles	302
Network Access	308
Amazon VPC	309
Benefits of Using a VPC	309
Differences Between EC2-Classic and EC2-VPC	322
Amazon VPC Documentation	311
Supported Platforms	312
Migrating from EC2-Classic to a VPC	314
Instance IP Addressing	320
Private Addresses and Internal DNS Hostnames	321

Public IP Addresses and External DNS Hostnames	321
Differences Between EC2-Classic and EC2-VPC	322
Determining Your Public, Private, and Elastic IP Addresses	323
Assigning a Public IP Address	324
Multiple Private IP Addresses	325
Elastic IP Addresses	329
Elastic IP Addresses in EC2-Classic	330
Elastic IP Addresses in a VPC	330
Differences Between EC2-Classic and EC2-VPC	331
Allocating an Elastic IP Address	331
Describing Your Elastic IP Addresses	332
Associating an Elastic IP Address with a Running Instance	332
Associating an Elastic IP Address with a Different Running Instance	333
Releasing an Elastic IP Address	333
Using Reverse DNS for Email Applications	334
Elastic IP Address Limit	334
Elastic Network Interfaces	334
Private IP Addresses Per ENI Per Instance Type	335
Creating a Management Network	337
Use Network and Security Appliances in Your VPC	337
Creating Dual-homed Instances with Workloads/Roles on Distinct Subnets	338
Create a Low Budget High Availability Solution	338
Best Practices for Configuring Network Interfaces	338
Creating a Network Interface	338
Deleting a Network Interface	339
Viewing Details about a Network Interface	339
Attaching a Network Interface When Launching an Instance	340
Attaching a Network Interface to a Stopped or Running Instance	341
Detaching a Network Interface from an Instance	342
Changing the Security Group of a Network Interface	342
Changing the Source/Destination Checking of a Network Interface	343
Associating an Elastic IP Address with a Network Interface	343
Disassociating an Elastic IP Address from a Network Interface	344
Changing Termination Behavior for a Network Interface	344
Adding or Editing a Description for a Network Interface	345
Adding or Editing Tags for a Network Interface	345
Enabling Enhanced Networking	346
Requirements	346
Testing Whether Enhanced Networking Is Enabled	347
Enabling Enhanced Networking on Windows	348
Storage	350
Amazon EBS	351
Features of Amazon EBS	352
EBS Volumes	353
EBS Snapshots	381
EBS Encryption	387
EBS Performance	389
API and Command Overview	401
Instance Store	403
Instance Storage Concepts	404
Instance Stores Available on Instance Types	405
Instance Store Device Names	406
Instance Store Usage Scenarios	407
Adding Instance Store Volumes to an AMI	408
Amazon S3	409
Amazon S3 and Amazon EC2	409
Block Device Mapping	411
Block Device Mapping Concepts	411

AMI Block Device Mapping	414
Instance Block Device Mapping	416
Using Public Data Sets	421
Public Data Set Concepts	421
Finding Public Data Sets	421
Creating a Public Data Set Volume from a Snapshot	422
Attaching and Mounting the Public Data Set Volume	423
Resources and Tags	424
Resource Locations	424
Listing and Filtering Your Resources	425
Listing Resources Using the Console	425
Filtering Resources Using the Console	426
Listing and Filtering Using the CLI and API	427
Tagging Your Resources	427
Tag Basics	428
Tag Restrictions	428
Tagging Your Resources for Billing	430
Working with Tags in the Console	430
API and CLI Overview	435
Service Limits	436
Viewing Your Current Limits	436
Requesting a Limit Increase	437
Usage Reports	437
Available Reports	437
Getting Set Up for Usage Reports	437
Granting IAM Users Access to the Amazon EC2 Usage Reports	439
Instance Usage	439
Reserved Instance Utilization	443
AWS Management Pack	449
Overview of AWS Management Pack for System Center 2012	450
Overview of AWS Management Pack for System Center 2007 R2	451
Downloading	452
Deploying	453
Step 1: Installing the AWS Management Pack	453
Step 2: Configuring the Watcher Node	455
Step 3: Create an AWS Run As Account	455
Step 4: Run the Add Monitoring Wizard	458
Using	462
Views	462
Discoveries	472
Monitors	474
Rules	475
Events	478
Health Model	479
Customizing the AWS Management Pack	480
Troubleshooting	480
AWS Diagnostics for Microsoft Windows Server	482
Analysis Rules	482
Analyzing the Current Instance	483
Collecting Data From an Offline Instance	485
Data File Storage	485
Troubleshooting	487
No console output	487
Instance terminates immediately	488
"Password is not available"	488
"Password not available yet"	489
"Cannot retrieve Windows password"	489
"Waiting for the metadata service"	489

Remote Desktop can't connect to the remote computer	492
RDP displays a black screen instead of the desktop	494
"Unable to activate Windows"	495
"Windows is not genuine (0x80070005)"	495
"No Terminal Server License Servers available to provide a license"	496
Instance loses network connectivity or scheduled tasks don't run when expected	496
Document History	497
AWS Glossary	508

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. 74)
- [Tags](#) (p. 427)

Networking and Security

- [Amazon EC2 Key Pairs](#) (p. 260)
- [Security Groups](#) (p. 264)
- [Elastic IP Addresses \(EIP\)](#) (p. 329)
- [Amazon EC2 and Amazon VPC](#) (p. 309)

Storage

- [Amazon EBS](#) (p. 351)
- [Instance Store](#) (p. 403)

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 Web Application Hosting for Microsoft Windows](#)

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 Elastic Compute Cloud 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 Elastic Compute Cloud 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 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 Usage 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

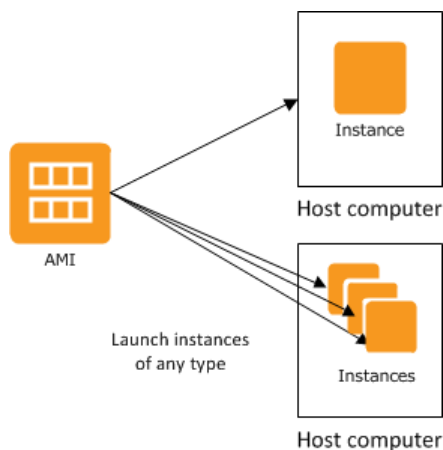
- [Amazon Machine Images and Instances \(p. 5\)](#)
- [Regions and Availability Zones \(p. 6\)](#)
- [Storage \(p. 6\)](#)
- [Root Device Volume \(p. 8\)](#)
- [Networking and Security \(p. 10\)](#)
- [AWS Identity and Access Management \(p. 10\)](#)
- [Differences between Windows Server and an Amazon 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 [Instance Type Details](#). You can also launch multiple instances from an AMI, as shown in the following figure.



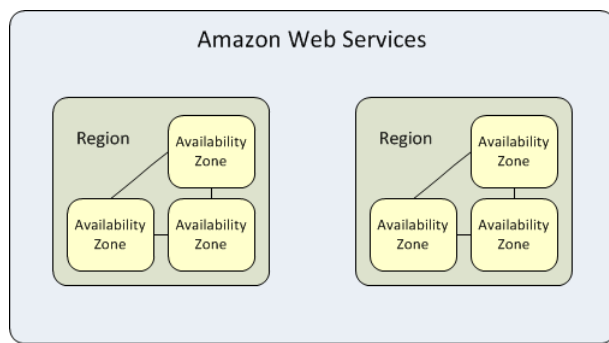
Your Windows instances keep running until you stop or terminate them, or until they fail. If an instance fails, you can launch a new one from the AMI.

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.



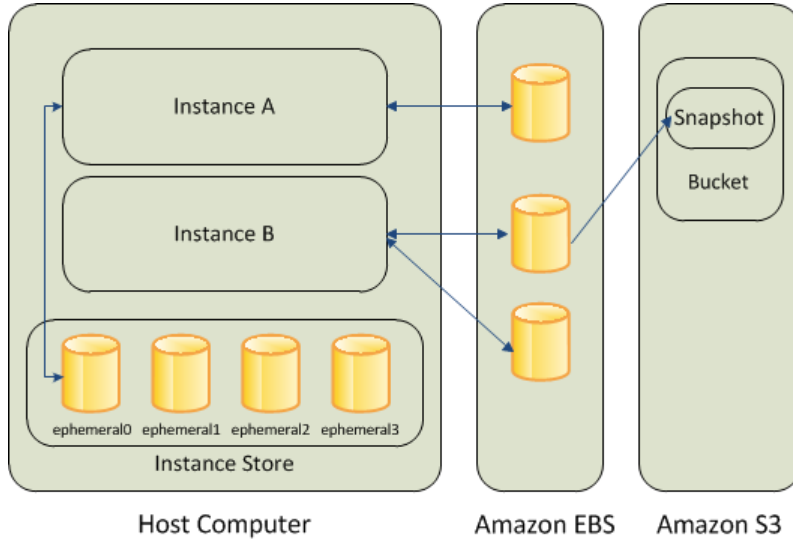
For more information about the available regions and Availability Zones, see [Using Regions and Availability Zones](#) in the *Amazon Elastic Compute Cloud User Guide for Linux*.

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 \(p. 403\)](#)
- [Amazon Simple Storage Service \(Amazon S3\)](#)

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

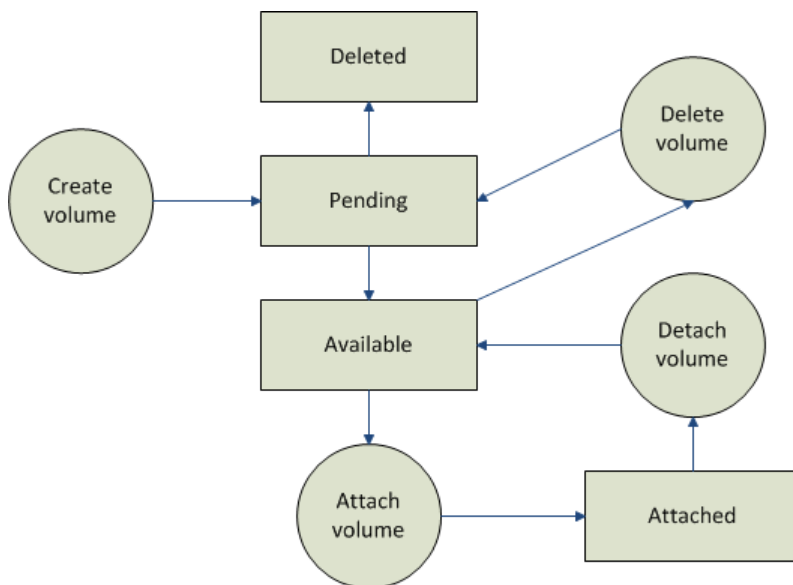


Amazon EBS Volumes

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. 351).

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 \(p. 403\)](#).

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 \(p. 48\)](#).

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. 312).

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. 320).

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](#) (p. 264).

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. 273)
- [IAM Policies for Amazon EC2](#) (p. 274)
- [IAM Roles for Amazon EC2](#) (p. 302)
- [Identity and Access Management \(IAM\)](#)
- [Using IAM](#)

Differences between Windows Server and an Amazon EC2 Windows Instance

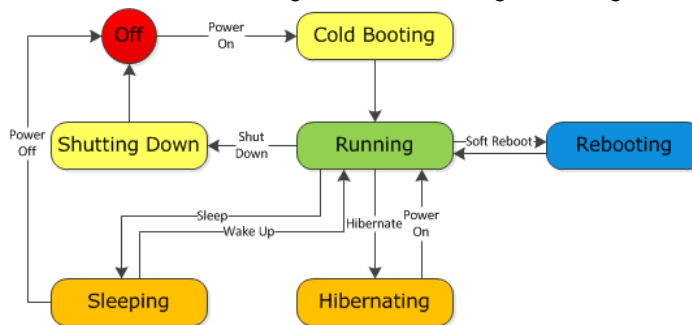
After you launch your Amazon EC2 Windows instance, it behaves like a traditional server running Windows Server. For example, both Windows Server and an Amazon EC2 instance can be used to run your web applications, conduct batch processing, or manage applications requiring large-scale computations. However, 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.

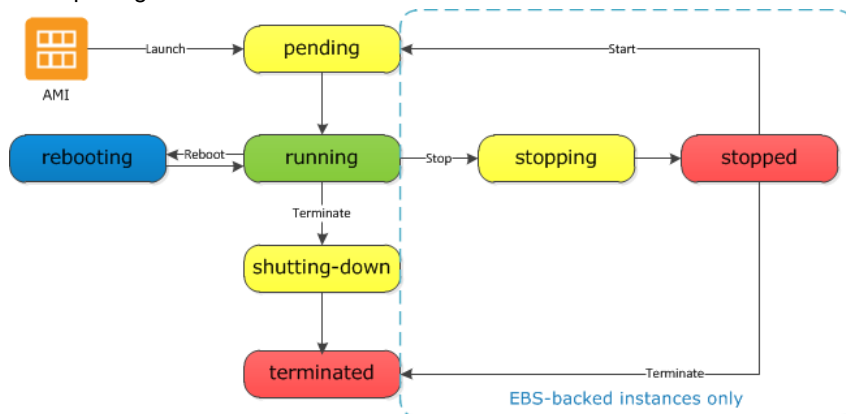
Windows Server	Amazon EC2 Windows Instance
Resources and capacity are physically limited.	Resources and capacity are scalable.
You pay for the infrastructure, even if you don't use it.	You pay for the usage of the infrastructure. We stop charging you for the instance as soon as you stop or terminate 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 <i>cold booting</i>).	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.



**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows**
**Designing Your Applications to Run on Amazon EC2
Windows Instances**

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.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Designing Your Applications to Run on Amazon EC2
Windows Instances**

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 Usage 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. Go to <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.

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 **Select Policy Template** section, click **Select** next to the **Administrator Access** policy template.
5. Click **Next Step** and then click **Create Group**.

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

To create the IAM user, 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 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. In the content pane, under **User Name**, click the name of the user you just created. (You might need to scroll down to find the user in the list.)
5. In the content pane, in the **Groups** section, click **Add User to Groups**.
6. Select the **Administrators** group and then click **Add to Groups**.
7. In the content pane, in the **Security Credentials** section (you might need to scroll down to find this section), 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**.

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 **Create Account Alias** 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. 272).

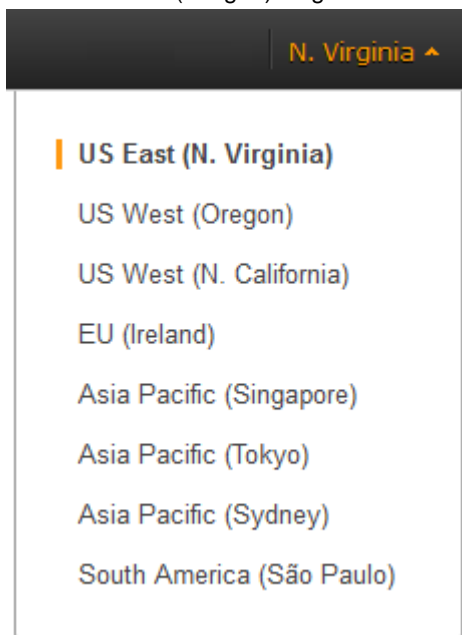
Create a Key Pair

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. 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 \(p. 260\)](#).

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 \(p. 312\)](#). 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 Virtual Private Cloud 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 \(p. 6\)](#).

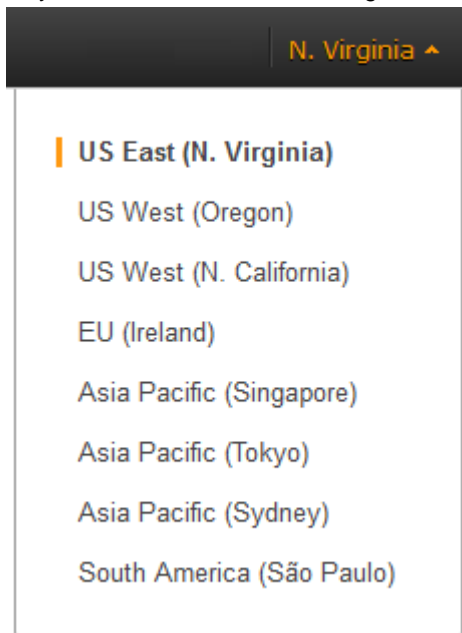
Tip

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.
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, ensure that your default VPC is selected; it's 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.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Create a Security Group**

For more information, see [Amazon EC2 Security Groups](#) (p. 264).

Getting Started with Amazon EC2 Windows Instances

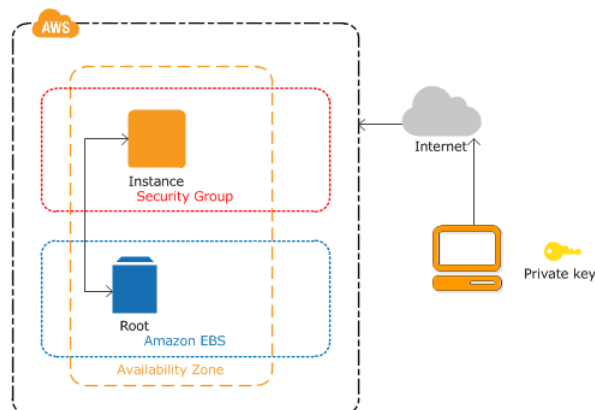
This tutorial provides a hands-on introduction to using Amazon EC2 using the AWS Management Console, a point-and-click web-based interface. We'll launch and connect to a Windows instance.

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 \(p. 138\)](#)

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

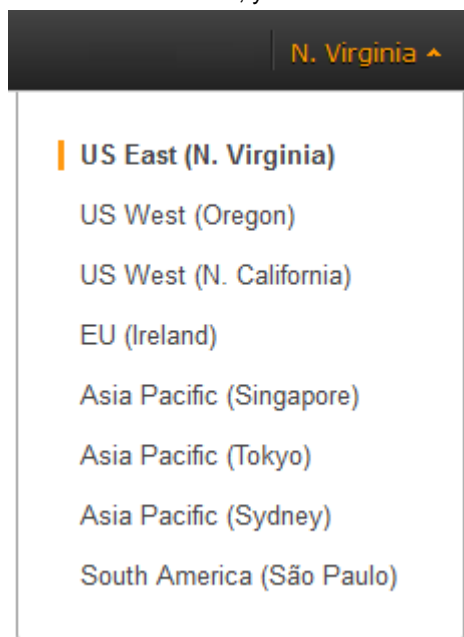
If you'd prefer to launch a Linux instance, see this tutorial in the *Amazon Elastic Compute Cloud User Guide for Linux*: [Getting Started with Amazon EC2 Linux Instances](#).

Launch a Windows Instance

You can launch a Windows instance using the AWS Management Console as described following. 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.
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, click **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 for your instance. The **t2.micro** instance type is selected by default. Alternatively, select **All generations** from the filter list, and then select the `t1.micro` instance type. Note that these are the only instance types eligible for the free tier.

Note

[T2 instances](#) (p. 76) must be launched into a VPC. If your AWS account supports EC2-Classic and you do not have any VPCs, the launch wizard creates a VPC for you. Otherwise,

if you have one or more VPCs, click **Next: Configure Instance Details** to select a VPC and subnet.

6. Click **Review and Launch** to let the wizard complete the other configuration settings for you.
7. On the **Review Instance Launch** page, you can review the settings for your instance.

Under **Security Groups**, you'll see that the wizard created and selected a security group for you. The security group includes basic firewall rules that enable you to connect to your instance. For a Windows instance, you connect through Remote Desktop Protocol (RDP) on port 3389.

Caution

The security group the wizard authorizes all IP addresses to access your instance over the specified ports (for example, RDP). This is acceptable for the short exercise in this tutorial, but it's unsafe for production environments. In production, you'll authorize only a specific IP address or range of IP addresses to access your instance.

If you have an existing security group you'd prefer to use, you can click **Edit security groups**, and select your group on the **Configure Security Group** page. When done, click **Review and Launch** to return to the **Review Instance Launch** page.

8. Click **Launch**.
9. 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.

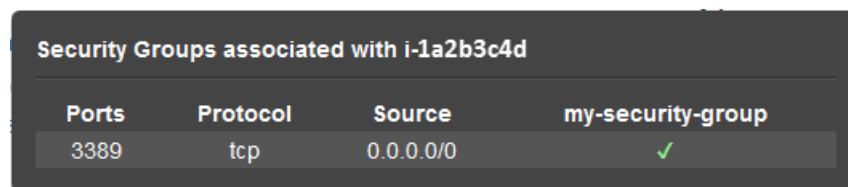
Alternatively, you can create a new key pair. Select **Create a new key pair**, enter a name for the key pair, and then click **Download Key Pair**. This is the only chance for you to save the private key file, so be sure to download it. Save the private key file in a safe place. You'll need to provide 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 click **Launch Instances**.

10. A confirmation page lets you know that your instance is launching. Click **View Instances** to close the confirmation page and return to the console.
11. 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, click the Show/Hide icon in the top right corner of the **Instances** page and select **Public DNS**.)
12. Record the public DNS name for your instance because you'll need it for the next step.
13. (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 click **view rules**.



Security Groups associated with i-1a2b3c4d			
Ports	Protocol	Source	my-security-group
3389	tcp	0.0.0.0/0	✓

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

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](#).

Important

If you are connecting to a Windows 2012 R2 instance using Mac OS X, the Remote Desktop Connection client from the Microsoft website may not work. Use the Microsoft Remote Desktop app from the Apple iTunes store instead.

To connect to your Windows instance

1. In the Amazon EC2 console, select the instance, and then click **Connect**.
2. In the **Connect To Your Instance** dialog box, click **Get Password** (it will take a few minutes after the instance is launched before the password is available).
3. Click **Browse** and navigate to the private key file you created when you launched the instance. Select the file and click **Open** to copy the entire contents of the file into contents box.
4. Click **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. Click **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 click **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 double-click 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, click **Connect** to connect to your instance. If you are using **Microsoft Remote Desktop** on a Mac, skip the next step.
8. Log in to the instance as prompted, using the default **Administrator** account and the default administrator password that you recorded or copied previously. If your **Remote Desktop Connection** already has an Administrator account set up, you may need to click 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 click **Yes** or **Continue** to continue if you trust the certificate.
 - a. If you are using **Remote Desktop Connection** from a Windows PC, click **View certificate**. If you are using **Microsoft Remote Desktop** on a Mac, click **Show Certificate**.
 - b. Click 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, click **Actions**, and then click **Get System Log**.
- d. In the system log output, look for an entry labelled `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 click **OK**. If you are using **Microsoft Remote Desktop** on a Mac, return to the **Verify Certificate** and click **Continue**.
- f. If you are using **Remote Desktop Connection** from a Windows PC, click **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 Amazon 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, click **Alarms**.
4. Click **Create Alarm**, and then in the **CloudWatch Metrics by Category** pane, select **EC2 Metrics**.
5. Select a metric using the following procedure, and then click **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**.

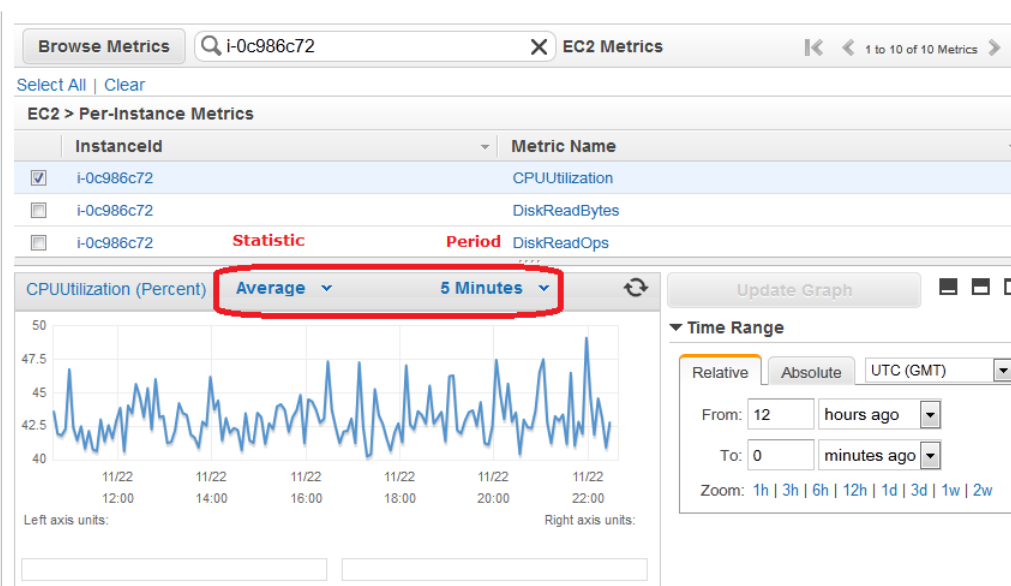
Amazon Elastic Compute Cloud User Guide for Microsoft Windows
Create a CloudWatch Alarm to Monitor Your Instance

1. Select Metric
2. Define Alarm

Back Next

Cancel

To create an alarm, first select a metric by browsing or searching on the right. Once you find the metric you want, select it and then click Next.



6. Define the alarm using the following procedure, and then click **Create Alarm**:
- Under **Alarm Threshold**, in the **Name** box, enter a unique name for the alarm, for example: **myTestAlarm**.
 - In the **Description** field, enter a description of the alarm, for example: **CPU usage is lower than 70 percent**.
 - Under **Whenever**, next to **is**, select **<** from the list and enter 70 in the box.
 - Under **Whenever**, next to **for**, enter 5 in the box.

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

- Under **Actions**, in the **Whenever this alarm** drop-down list, select **State is ALARM**.
- In the **Send notification to** list, select an existing Amazon SNS topic or create a new one. To create a new Amazon SNS topic, click **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.

The screenshot displays the 'Alarm Threshold' configuration page in the Amazon CloudWatch console. On the left, a navigation pane shows '1. Select Metric' and '2. Define Alarm'. Below it are 'Back', 'Next', and 'Cancel' buttons, along with a 'Create Alarm' button. A message box prompts the user to set the alarm threshold, actions, and notification, and click 'Create Alarm'. The main area is titled 'Alarm Threshold' and contains the following fields: 'Name' (myTestAlarm), 'Description' (CPU usage is lower than 70 percent), 'Whenever' (CPUUtilization), 'is' (less than 70), and 'for' (5 consecutive period(s)). Below this is the 'Actions' section, which includes a 'Notification' box with 'Whenever this alarm' set to 'State is ALARM' and 'Send notification to' set to 'Please select an SNS topic'. On the right, the 'Alarm Preview' section shows a line graph of CPUUtilization over time, with a red threshold line at 70%. Below the graph are fields for 'Namespace' (AWS/EC2), 'InstancedId' (i-0c986c72), 'Metric Name' (CPUUtilization), 'Period' (5 Minutes), and 'Statistic' (Average).

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 Free Usage 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 Amazon CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, click **Alarms**.
3. In the alarms list, select the alarm you created, and then click **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 [Free Usage 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, click **Instances**. In the list of instances, locate the instance you want to terminate.
2. Right-click the instance, and then click **Terminate**.
3. Click **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 *Using IAM* guide.
- Implement the least permissive rules for your security group. For more information, see [Security Group Rules \(p. 265\)](#).
- Regularly patch, update, and secure the operating system and applications on your instance. For more information about updating Windows Server, go to [Windows Server Update Services](#) on the Microsoft website.
- 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 \(VPC\) \(p. 309\)](#).

Storage

- Understand the implications of the root device type for data persistence, backup, and recovery. For more information, see [Storage for the Root Device \(p. 48\)](#).
- Use separate Amazon EBS volumes for the operating system versus your data. Ensure that the volume with your data persists after instance termination.
- 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 \(p. 100\)](#) and [Tagging Your Amazon EC2 Resources \(p. 427\)](#).
- 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 \(p. 436\)](#).

Backup and Recovery

- Regularly back up your instance using [Amazon EBS snapshots \(p. 381\)](#) 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 \(p. 320\)](#).
- Monitor and respond to events. For more information, see [Monitoring Amazon EC2 \(p. 186\)](#).
- 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 \(ENI\) \(p. 334\)](#). 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.

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

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

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 Elastic Compute Cloud User Guide for Linux*.

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 to 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=1/|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-Es7Q10-bp28EKv' );
define( 'LOGGED_IN_SALT' , ' ;j{00P*owZf)kVD+FVLn~> .|Y%Ug4#I^*LVd9QeZ^&XmK|e(76miC+&N&+^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 \(p. 97\)](#). 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 a Volume \(p. 377\)](#)). 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: 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 \(p. 35\)](#)
- [Task 2: Configure Your Head Node \(p. 37\)](#)
- [Task 3: Set Up the Compute Node \(p. 39\)](#)
- [Task 4: Scale Your HPC Compute Nodes \(Optional\) \(p. 41\)](#)

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 Elastic Compute Cloud 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 \(p. 42\)](#) 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 \(p. 43\)](#) 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" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.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, 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
```

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Create-HPC-sec-group.bat**

```
:: 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
```

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Create-HPC-sec-group.bat**

```
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)

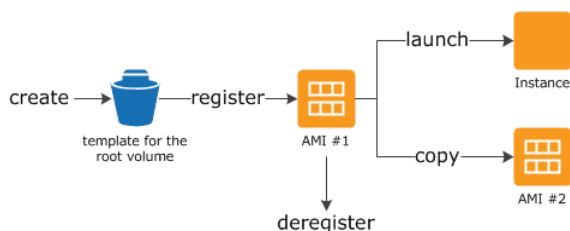
An Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud. You specify an AMI when you launch an instance, and you can launch as many instances from the AMI as you need.

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. 48) and [Finding an AMI](#) (p. 51).

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. 74).

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 \(p. 62\)](#). To create an instance store-backed AMI, see [Creating an Instance Store-Backed Windows AMI \(p. 64\)](#).

To help categorize and manage your AMIs, you can assign custom *tags* to them. For more information, see [Tagging Your Amazon EC2 Resources \(p. 427\)](#).

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 \(p. 53\)](#).

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 \(p. 58\)](#).

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 \(p. 72\)](#).

AWS Windows AMIs

Amazon Web Services (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; then launch an instance using that AMI. You connect to the instance using Remote Desktop Connection, just as you would with any other Windows server.

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)

AWS also provides a set of publicly available AMIs that include SQL Server, SQL Server Express, Internet Information Services (IIS), and ASP.NET to help you get started quickly. You can use one or more of these AMIs to deploy your applications. For example, you can use an AWS Windows AMI with SQL Server Express, IIS, and ASP.NET to launch an instance that runs web and ASP.NET applications. Launching an instance from an AWS Windows AMI with SQL Server offers you the flexibility to run the instance as a database server. Or you can launch an instance from one of the basic Windows AMIs, customize the instance by installing the software and applications of your choice, and then register the customized instance as an AMI. You can then use this customized AMI to launch additional instances that include your chosen software and applications.

To locate the Windows AMIs provided by AWS using the Amazon EC2 console, see [Windows AMIs](#).

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 an AMI \(p. 51\)](#).

You can also create an AMI from your own Windows computer. For more information, see [Importing and Exporting Instances \(p. 107\)](#).

Update Schedule

AWS provides updated, full patched Windows AMIs within five business days of Microsoft's patch Tuesday (the second Tuesday of each month). The new Windows AMIs have new AMI IDs. To find the latest Windows AMIs, use the AMI name instead of the AMI ID. The basic structure of the AMI name is usually the same, with a new date added to the end. You can use a query or script to search for an AMI by name, confirm that you've found the correct AMI, and then launch your instance. AWS removes the previously published Windows AMIs within 10 business days after publishing updated Windows AMIs.

Configuration Settings

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

An Amazon EC2 Windows AMI comes 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 \(p. 152\)](#).

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 AMIs. Instances launched from your customized AMI are launched with the new configuration.

Xen Drivers

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 [Xen Drivers \(p. 165\)](#).

Keeping Your Instances Up-to-Date

At their initial launch, your Windows instances contain all the latest security updates. However, after you launch an instance, you are responsible for managing future updates, including the updates issued after

you built the AMI. You can use the Windows Update service, or the Automatic Updates tool available on your instance to deploy the Microsoft updates. Any third-party software you deploy must also be kept up-to-date using whatever mechanisms are appropriate for that software. We recommend that you run the Windows Update service as a first step after every Windows instance that you launch.

You can reboot a Windows instance after installing updates. For more information, see [Reboot Your Instance](#) (p. 143).

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. 48)
- [Storage for the Root Device](#) (p. 48)

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. 53). Developers can charge for their AMIs. For more information, see [Paid AMIs](#) (p. 58).

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

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Storage for the Root Device**

Characteristic	Amazon EBS-Backed	Amazon Instance Store-Backed
Size limit	1 TiB	10 GiB
Root device volume	Amazon EBS volume	Instance store volume
Data persistence	Data on Amazon EBS volumes persists after instance termination*; you can also attach instance store volumes that don't persist after instance termination.	Data on instance store volumes persists only during the life of the instance; you can also attach Amazon EBS volumes that persist after instance termination.
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.
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`, which causes the volume to be deleted upon instance termination. For information about how to change this so that the volume persists following 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 \(p. 140\)](#).

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 \(p. 411\)](#). For information about what happens to the instance store volumes when you stop an instance, see [Stop and Start Your Instance \(p. 140\)](#).

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 \(p. 62\)](#).

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 an 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
- 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

Finding a Windows AMI Using the Amazon EC2 Console

The Amazon EC2 console provides one way to see available Windows AMIs.

To find a Windows AMI using the console

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 the AMIs that interest you. For example, to list all Windows AMIs provided by AWS, select **Public images**, **Amazon images**, and then **Windows** from the **Filter** lists.
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. Before you select an AMI, it's important that you check whether it's backed by instance store or by Amazon EBS and that you are aware of the effects of this difference. For more information, see [Storage for the Root Device \(p. 48\)](#).
7. 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 \(p. 131\)](#). If you're not ready to launch the instance now, write down the AMI ID (ami-xxxxxxx) for later.

Finding an AMI Using the Command Line

You can use command line parameters to list only the types of AMIs that interest you. For example, the following commands find public AMIs owned by you or Amazon.

- [describe-images](#) (AWS CLI)

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

- [ec2-describe-images](#) (Amazon EC2 CLI)

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

Add the following filters to a command to display only Windows AMIs backed by Amazon EBS:

- [describe-images](#) (AWS CLI)

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

- [ec2-describe-images](#) (Amazon EC2 CLI)

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

- [Get-EC2Image](#) (AWS Tools for Windows PowerShell)

```
PS C:\> $platform_values = New-Object 'collections.generic.list[string]'  
PS C:\> $platform_values.add("windows")  
PS C:\> $filter_platform = New-Object Amazon.EC2.Model.Filter -Property  
@{Name="platform"; Values=$platform_values}  
PS C:\> $device_values = New-Object 'collections.generic.list[string]'  
PS C:\> $device_values.add("ebs")  
PS C:\> $filter_device = New-Object Amazon.EC2.Model.Filter -Property  
@{Name="root-device-type"; Values=$device_values}  
PS C:\> Get-EC2Image ... -Filter $filter_platform, $filter_device
```

After locating an AMI that meets your needs, write down its ID (ami-xxxxxxx). You can use this AMI to launch an instances. For more information, see one of the following:

- [Launching an Instance Using the AWS CLI](#) in the *AWS Command Line Interface User Guide*
- [Launching an Instance Using the Amazon EC2 CLI](#) in the *Amazon Elastic Compute Cloud Command Line Reference*

- [Launching an Instance Using Windows PowerShell](#) in the *AWS Tools for Windows PowerShell User Guide*

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.

Topics

- [Finding Shared AMIs](#) (p. 53)
- [Making an AMI Public](#) (p. 55)
- [Sharing an AMI with Specific AWS Accounts](#) (p. 56)
- [Using Bookmarks](#) (p. 58)

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 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.
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>#LaunchInstanceWizard: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#LaunchInstanceWizard: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.

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.

Topics

- [Selling Your AMI \(p. 59\)](#)
- [Finding a Paid AMI \(p. 59\)](#)
- [Purchase a Paid AMI \(p. 60\)](#)
- [Getting the Product Code for Your Instance \(p. 60\)](#)

- [Using Paid Support](#) (p. 61)
- [Bills for Paid and Supported AMIs](#) (p. 61)
- [Managing Your AWS Marketplace Subscriptions](#) (p. 61)

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, **Marketplace images** from the second **Filter** list, and the operating system from the third **Filter** 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 \(p. 136\)](#).

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 \(p. 136\)](#).

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 \(p. 100\)](#).

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, and select **Terminate** from the **Actions** menu. 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 Elastic Compute Cloud User Guide for Linux*.

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 \(p. 64\)](#).

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 \(p. 151\)](#).
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 \(p. 164\)](#).
 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**, and then click **Stop Instance**.
 5. On the **Instances** page of the Amazon EC2 console, select your instance. Click **Actions** 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 \(p. 72\)](#).

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.

If you need to create an instance store-backed Linux AMI, see [Creating an Instance Store-Backed Linux AMI](#) in the *Amazon Elastic Compute Cloud User Guide for Linux*.

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 \(p. 62\)](#).

Contents

- [Instance Store-Backed Windows AMIs \(p. 64\)](#)
- [Preparing to Create an Instance Store-Backed Windows AMI \(p. 65\)](#)
- [Bundling an Instance Store-Backed Windows Instance \(p. 66\)](#)
- [Registering an Instance Store-Backed Windows AMI \(p. 67\)](#)

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.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Preparing to Create an Instance Store-Backed Windows
AMI**

- 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 \(p. 350\)](#).

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 \(p. 151\)](#).
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 \(p. 164\)](#).

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.

Note

AMIs with encrypted volumes cannot be copied using the AWS Management Console. Instead, you must copy each volume's snapshot to the target region and create a new AMI in that region using the copied snapshots in the block device mappings.

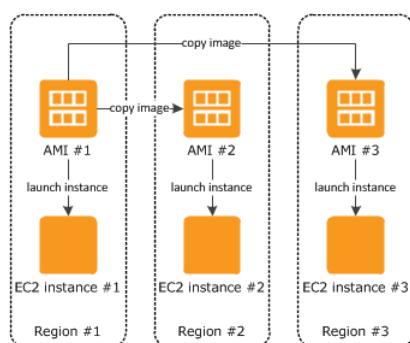
Copying your AMIs provides the following benefits:

- **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.

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.

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

Copying an Amazon EC2 AMI

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.

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

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**. If your AMI has encrypted volumes, see [Copying an Amazon EC2 AMI with Encrypted Volumes \(p. 69\)](#).

Note

AMIs with encrypted volumes cannot be copied using the AWS Management Console. Instead, you must copy each volume's snapshot to the target region and create a new AMI in that region using the copied snapshots in the block device mappings.

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)

Copying an Amazon EC2 AMI with Encrypted Volumes

AMIs with encrypted volumes cannot be copied using the AWS Management Console. However, you can manually copy the Amazon EBS volume snapshots from your source region to your destination region, and then register a new AMI in the destination region with the copied snapshots. This will give you the same end result as copying the AMI using the AWS Management Console. The following procedure demonstrated how to do this with the AWS CLI, but you can also use the Amazon EC2 CLI or the SDK tools if you prefer.

To copy an Amazon EC2 AMI with encrypted volumes using the AWS CLI

1. Collect the needed information for the AMI you would like to copy with the **describe-images** command. Take note of the following:

- VirtualizationType
- SrvioNetSupport
- BlockDeviceMappings, including the SnapshotId, VolumeType, and Encrypted values
- Architecture
- RootDeviceName

```
C:\> aws ec2 describe-images --region us-west-2 --image-id ami-1a2b3c4d
{
  "Images": [
    {
      "VirtualizationType": "hvm",
      "Name": "ec2-encrypted-volume-ami",
      "Hypervisor": "xen",
      "SrvioNetSupport": "simple",
      "ImageId": "ami-1a2b3c4d",
      "State": "available",
      "BlockDeviceMappings": [
        {
          "DeviceName": "/dev/xvda",
          "Ebs": {
            "DeleteOnTermination": true,
            "SnapshotId": "snap-2345bcde",
            "VolumeSize": 8,
            "VolumeType": "gp2",
            "Encrypted": false
          }
        },
        {
          "DeviceName": "/dev/sdb",
          "Ebs": {
            "DeleteOnTermination": false,
            "SnapshotId": "snap-3456cdef",
            "VolumeSize": 100,
            "VolumeType": "gp2",
            "Encrypted": false
          }
        },
        {
          "DeviceName": "/dev/sdc",
          "Ebs": {
            "DeleteOnTermination": false,
            "SnapshotId": "snap-abcd1234",
            "VolumeSize": 150,
            "VolumeType": "gp2",
            "Encrypted": true
          }
        }
      ],
      "Architecture": "x86_64",
      "ImageLocation": "012345678910/ec2-encrypted-volume-ami",
      "RootDeviceType": "ebs",
    }
  ]
}
```

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Stopping a Pending AMI Copy Operation**

```
        "OwnerId": "012345678910",
        "RootDeviceName": "/dev/xvda",
        "Public": false,
        "ImageType": "machine",
        "Description": "/dev/xvdc is encrypted"
    }
}
}
```

2. Copy each Amazon EBS snapshot contained in the AMI to the destination region for your new AMI. This can be done in the AWS Management Console or with the command line tools. It is helpful to note in the snapshot description which device the snapshot is for. This will help you configure the block device mapping on your new AMI later. The following command copies the snapshot associated with `/dev/sdc` on our AMI.

```
C:\> aws ec2 copy-snapshot --source-region us-west-2 --source-snapshot-id
snap-abcd1234 --destination-region us-east-1 --description "Copy of /dev/sdc
from ami-1a2b3c4d"
{
    "SnapshotId": "snap-4321dcba"
}
```

3. Register your new AMI with the copied snapshots in the block device mapping and the information you recorded earlier using the **register-image** command. For more information, see [register-image](#).

```
C:\> aws ec2 register-image --region us-east-1 --name "my-copied-ami" --
architecture x86_64 --root-device-name /dev/xvda --block-device-mappings
"[{"DeviceName": "/dev/xvda", "Ebs": {"DeleteOnTermination": true, "Snap
shotId": "snap-5432edcb", "VolumeType": "gp2"}}, {"Device
Name": "/dev/sdb", "Ebs": {"DeleteOnTermination": false, "Snapshot
Id": "snap-6543fedc", "VolumeType": "gp2"}}, {"Device
Name": "/dev/sdc", "Ebs": {"DeleteOnTermination": false, "Snapshot
Id": "snap-4321dcba", "VolumeType": "gp2"}}]" --virtualization-type
hvm --sriov-net-support simple
{
    "ImageId": "ami-1d2c3b4a"
}
```

4. (Optional) Run the **describe-images** command on your new AMI and compare it to the original to ensure that everything is correct. If not, you can deregister the image, make your corrections, and try registering the AMI again. For more information, see [deregister-image](#).

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.

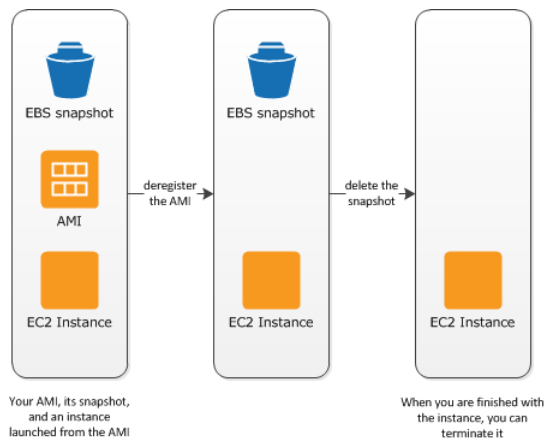
Topics

- [Cleaning Up Your Amazon EBS-Backed AMI \(p. 72\)](#)
- [Cleaning Up Your Instance Store-Backed AMI \(p. 73\)](#)

Cleaning Up Your Amazon EBS-Backed AMI

When you deregister an Amazon EBS-backed AMI, it doesn't affect the snapshot that we created when you created the AMI. You'll continue to incur usage costs for this snapshot in Amazon EBS. 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**.

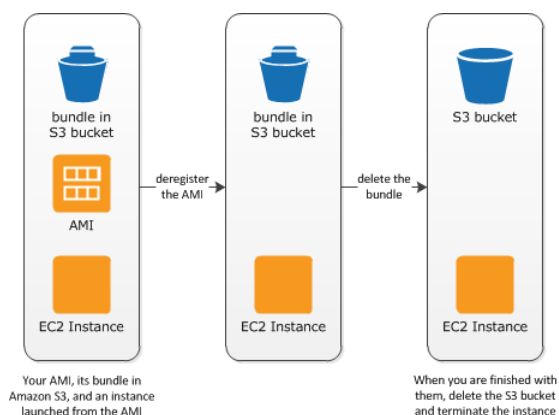
The AMI status is now `unavailable`.

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**.

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 \(p. 126\)](#)

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 \(p. 74\)](#).

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 \(p. 48\)](#).

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 \(p. 312\)](#) and [Amazon EC2 and Amazon Virtual Private Cloud \(VPC\) \(p. 309\)](#).

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.

The maximum transmission unit (MTU) for an instance depends on its instance type. The following instance types provide 9001 MTU (jumbo frames): CC2, C3, R3, CG1, CR1, G2, HS1, HI1, I2, T2, and M3. The other instance types provide 1500 MTU (Ethernet v2 frames).

To obtain additional, dedicated capacity for Amazon EBS I/O, you can launch some instance types as Amazon EBS-optimized instances. For more information, see [Amazon EBS-Optimized Instances \(p. 93\)](#).

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 \(p. 94\)](#).

Available Instance Types

Amazon EC2 provides the current and previous generation instance types listed in the following tables.

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, see [How many instances can I run in Amazon EC2?](#)

Current Generation Instances

For the best performance, we recommend that you use current generation instance types and HVM AMIs when you launch new instances. For more information on current generation instance types, see the [Amazon EC2 Instances](#) detail page.

Instance Family	Current Generation 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 c3.8xlarge
Memory optimized	r3.large r3.xlarge r3.2xlarge r3.4xlarge r3.8xlarge
Storage optimized	i2.xlarge i2.2xlarge i2.4xlarge i2.8xlarge hs1.8xlarge
GPU instances	g2.2xlarge

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 after new instances launch. If you are currently using a previous generation instance and would like to see which one would be a suitable upgrade, see [Upgrade Paths](#).

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
GPU instances	cg1.4xlarge
Micro instances	t1.micro

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Instance Type Details](#).

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. If your needs change, you can resize your instance later on. For more information, see [Resizing Your Instance \(p. 97\)](#).

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 [Instance Type Details](#).

T2 instances are currently available in three instance sizes: `t2.micro`, `t2.small`, and `t2.medium`. Customers can launch T2 instances using the AWS management console, Amazon EC2 command line interface, and the AWS SDKs. T2 instances are available as On-Demand or Reserved Instances, but they cannot be purchased as Spot Instances. For more information, see [Amazon EC2 Instance Purchasing Options](#). T2 instance types are available as Amazon EBS-backed instances only.

Topics

- [Hardware Specifications \(p. 76\)](#)
- [CPU Credits \(p. 77\)](#)
- [EC2-VPC-only Support \(p. 78\)](#)
- [HVM AMI Support \(p. 78\)](#)
- [Default T2 Instance Limits \(p. 79\)](#)
- [Monitoring Your CPU Credits \(p. 79\)](#)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Instance Type Details](#).

CPU Credits

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.





Each T2 instance starts with a healthy initial CPU credit balance and then continuously (at a millisecond-level resolution) receives a set amount of CPU credits per hour, depending on instance size. 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 instances, for example) running at 25% utilization for two minutes. 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 maintaining a steady workload to avoid 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) rolls over to the next hour, 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 and the better it will perform when more performance is needed.

For example, a `t2.small` instance continuously receives credits at a rate of 12 CPU credits per hour. This capability provides a baseline performance equivalent to 20 percent of a CPU core. If at any moment the instance does not need all of the credits it receives, it stores the unused credits in its CPU credit balance for up to 24 hours. If this instance had a CPU utilization of 5% for the hour, it would have used 3 CPU minutes, but it would have earned 12 CPU minutes during the hour, so the difference of 9 CPU minutes would be added to the CPU credit balance.

If and when your `t2.small` instance needs to burst to more than 20 percent of a core, it draws from its CPU credit balance. Over time, if you find that your workload needs more CPU credits, or that your instance does not maintain a positive CPU credit balance, we recommend using a larger T2 size, such as the `t2.medium`, or upgrading to a fixed performance instance type.

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 CPU credit balance that an instance can accrue.

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

* There are limits to how many of your T2 instances will launch or start with the initial credit, which by default is set to 100 launches or starts per 24-hour period by 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](#).

** `t2.medium` instances have two vCPUs. The base performance is an aggregate of the two vCPUs; this can be 40% utilization on one vCPU, 20% each on two vCPUs, or any combination that does not exceed 40%.

The `t2.micro` and `t2.small` instance types launch with an initial balance of 30 CPU credits, and the `t2.medium` instance type launches with 60 CPU credits. This initial credit balance is designed to provide a good startup experience. The maximum 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 CPU credit balance of 144 CPU 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.

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.

EC2-VPC-only Support

T2 instances must be launched into an Amazon 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 EC2-Classic instance to a T2 instance type. For more information on EC2-Classic and EC2-VPC, see [Supported Platforms \(p. 312\)](#).

If your account supports EC2-Classic and you have not created any VPCs, you can do one of the following to launch a T2 instance:

- Create a VPC, and launch your instance into it. For more information, see [Getting Started with Amazon VPC](#) in the *Amazon Virtual Private Cloud Getting Started Guide*.
- Launch a T2 instance using the launch wizard in the Amazon EC2 console. The wizard creates a nondefault VPC in your account with the following attributes:
 - A subnet in each Availability Zone. By default, the wizard selects the subnet in the first Availability Zone in which to launch your instance. The public IP addressing attribute of each subnet is set to `true` so that instances launched into each subnet receive a public IP address. For more information, see [Modifying Your Subnet's Public IP Addressing Behavior](#) in the *Amazon Virtual Private Cloud User Guide*.
 - An Internet gateway, and a main route table that routes the VPC's traffic to the Internet gateway. This enables your VPC (and instances in your subnet) to communicate over the Internet. For more information about Internet gateways, see [Adding an Internet Gateway to Your VPC](#).
 - A default network ACL associated with all subnets, and a default security group. For more information about network ACLs, see [Network ACLs](#). For more information about security groups, see [Security Groups for Your VPC](#).

If you are using the Amazon EC2 API, the Amazon EC2 CLI, or the AWS CLI, you must have a default VPC in which to launch your T2 instance, or you must specify a subnet ID or network interface ID in the request.

By launching an instance into a VPC, you can leverage a number of features that are available only on the EC2-VPC platform; such as assigning multiple private IP addresses to your instances, or changing your instances' security groups. For more information about the benefits of using a VPC, see [Amazon EC2 and Amazon Virtual Private Cloud \(VPC\) \(p. 309\)](#). You can take steps to migrate your resources from EC2-Classic to EC2-VPC. For more information, see [Migrating from EC2-Classic to a VPC \(p. 314\)](#).

HVM AMI Support

T2 instances require HVM AMIs.

Default T2 Instance Limits

We limit the number of each T2 instance type that you can run simultaneously to 20. If you need more than 20 T2 instances, you can request more by using the [Amazon EC2 Instance Request Form](#).

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 \(p. 201\)](#).

Metric	Description
<code>CPUCreditUsage</code>	<p>(Only valid for T2 instances) The number of CPU credits consumed during the specified period. This includes both credits earned during the period and credits withdrawn from the CPU credit balance for bursting.</p> <p>This metric identifies the amount of time during which physical CPUs were used for processing instructions by virtual CPUs allocated to the instance.</p> <p>Note CPU Credit metrics are available at a 5 minute frequency.</p> <p>Units: Count</p>
<code>CPUCreditBalance</code>	<p>(Only valid for T2 instances) The number of CPU credits that an instance has accumulated.</p> <p>This metric is used to determine how long an instance can burst beyond its baseline performance level at a given rate.</p> <p>Note CPU Credit metrics are available at a 5 minute frequency.</p> <p>Units: Count</p>

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

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 \(p. 94\)](#).

You can enable enhanced networking capabilities for I2 instances. For more information, see [Enabling Enhanced Networking on Windows Instances in a VPC \(p. 346\)](#).

Topics

- [Hardware Specifications \(p. 80\)](#)
- [I2 Instance Limitations \(p. 80\)](#)
- [SSD Storage \(p. 80\)](#)
- [SSD I/O Performance \(p. 80\)](#)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Instance Type Details](#).

I2 Instance Limitations

We limit the number of I2 instances that you can run. If you need more I2 instances than the default limits described in the following table, you can request more I2 instances using the [Amazon EC2 Instance Request Form](#).

Instance Size	Default Instance Limit
i2.xlarge	8
i2.2xlarge	8
i2.4xlarge	4
i2.8xlarge	2

SSD Storage

The primary data storage for I2 instances is SSD-based instance storage. Like all instance storage, these volumes persist only for the life of the instance. When you terminate an instance, the applications and data in its instance store is erased. When you use an Amazon EBS-backed AMI, you can start and stop your instance. However, when you stop an instance, the data stored in the Amazon EBS volume persists, but data in the instance store volumes doesn't persist. We recommend that you regularly back up or replicate the data you've stored in instance storage.

For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 403\)](#).

SSD I/O Performance

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.

Note

Windows instances do not yet support TRIM.

H1 Instances

H1 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 H1 instances in a placement group. For more information, see [Placement Groups \(p. 94\)](#).

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](#).

Topics

- [Hardware Specifications \(p. 81\)](#)
- [Disk I/O Performance \(p. 81\)](#)
- [SSD Storage \(p. 81\)](#)

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 [Instance Type Details](#).

Disk I/O Performance

H1 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

This section contains important information you need to know about SSD storage. With SSD storage:

- 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 \(p. 403\)](#).

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 depth (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 TRIM support, use I2 instances. For more information, see [I2 Instances \(p. 79\)](#).

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 \(p. 94\)](#).

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](#).

Topics

- [Hardware Specifications \(p. 83\)](#)
- [Disk I/O Performance \(p. 83\)](#)

- [Storage Information \(p. 83\)](#)

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 are capable of delivering 2.4 GB per second of sequential read and 2.6 GB per second of sequential write performance when using a block size of 2 MiB.

HS1 instances do not currently support Amazon EBS optimization, but 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 [Instance Type Details](#).

Disk I/O Performance

HS1 instances are capable of delivering 2.6 GB per second of sequential read and write performance when using a block size of 2 MiB.

Storage Information

This section contains important information you need to know about the storage used with HS1 instances.

Instance Store with HS1 Instances

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. Therefore, when you stop an instance (when using an Amazon EBS-backed root volume), 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 \(p. 403\)](#).

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.

R3 Instances

R3 instances are optimized to deliver high memory performance and high sustainable bandwidth. They are well suited for the following scenarios:

- Relational databases and NoSQL databases (for example, MongoDB)
- In-memory analytics
- Memcache/Redis applications (for example, Elasticache)

You can cluster R3 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 \(p. 94\)](#).

You can enable enhanced networking capabilities for R3 instances. For more information, see [Enabling Enhanced Networking on Windows Instances in a VPC \(p. 346\)](#).

Topics

- [Hardware Specifications](#) (p. 84)
- [HVM AMI Support](#) (p. 84)
- [Default R3 Instance Limits](#) (p. 84)
- [SSD Storage](#) (p. 84)
- [SSD I/O Performance](#) (p. 84)
- [TRIM Support](#) (p. 85)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Instance Type Details](#).

HVM AMI Support

R3 instances have high-memory (up to 244 GiB of RAM), and require 64-bit operating systems to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on high-memory instance types. For these reasons, R3 instances support 64-bit HVM AMIs only. In addition, HVM AMIs are required to leverage the benefits of enhanced networking.

Default R3 Instance Limits

We limit the number of R3 instances that you can run simultaneously. If you need more R3 instances than the default limits described in the following table, you can request more by using the [Amazon EC2 Instance Request Form](#).

Instance Type	Default Instance Limit
r3.large	20
r3.xlarge	20
r3.2xlarge	20
r3.4xlarge	10
r3.8xlarge	5

SSD Storage

The primary data storage for R3 instances is SSD-based instance storage. Like all instance storage, these volumes persist only for the life of the instance. When you terminate an instance, the applications and data in its instance store are erased. When you use an Amazon EBS-backed AMI, you have the option to stop your instance and restart it later; however, when you stop an instance, the data stored in the Amazon EBS volumes persist, but data in the instance store volumes is lost. We recommend that you regularly back up or replicate the data you've stored in instance storage.

For more information about instance store volumes, see [Amazon EC2 Instance Store](#) (p. 403).

SSD I/O Performance

The largest R3 instances (r3.8xlarge) are capable of providing up to 150,000 4 kilobyte (KB) random read IOPS and up to 130,000 4 KB random first write IOPS.

TRIM Support

Windows instances do not yet support TRIM.

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 \(p. 94\)](#).

Topics

- [Hardware Specifications \(p. 85\)](#)
- [GPU Instance Limitations \(p. 85\)](#)
- [AMIs for GPU Instances \(p. 85\)](#)
- [Installing the NVIDIA Driver on Windows \(p. 86\)](#)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Instance Type Details](#).

GPU Instance Limitations

GPU instances currently have the following limitations:

- They aren't available in every region.
- They must be launched from HVM AMIs.
- They can't access the GPU unless the NVIDIA drivers are installed.
- We limit 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 a CG1 instance using any HVM AMI.

You can launch a G2 instance using Windows Server 2012 and Windows Server 2008 R2 AMIs. If you encounter the following error when launching a G2 instance, contact [Customer Service](#) or reach out through the [Amazon EC2 forum](#).

```
Client.UnsupportedOperation: Instances of type 'g2.2xlarge' may not be launched from AMI <ami-id>.
```

After you launch a G2 instance, you can create your own AMI from the instance. However, if you create a snapshot of the root volume of the instance, register it as an AMI, and then launch a G2 instance, you'll get the `Client.UnsupportedOperation` error. To launch a G2 instance from your own AMI, you must create the AMI from a G2 instance using the console (select the instance, click **Actions**, and then click **Create Image**), [create-image](#) (AWS CLI), or [ec2-create-image](#) (Amazon EC2 CLI).

Installing the NVIDIA Driver on Windows

To install the NVIDIA driver on your Windows instance, log on to your instance as 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.

When using Remote Desktop, GPUs that use the WDDM driver model are replaced with a non-accelerated Remote Desktop display driver. In order 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.

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 \(p. 76\)](#).

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

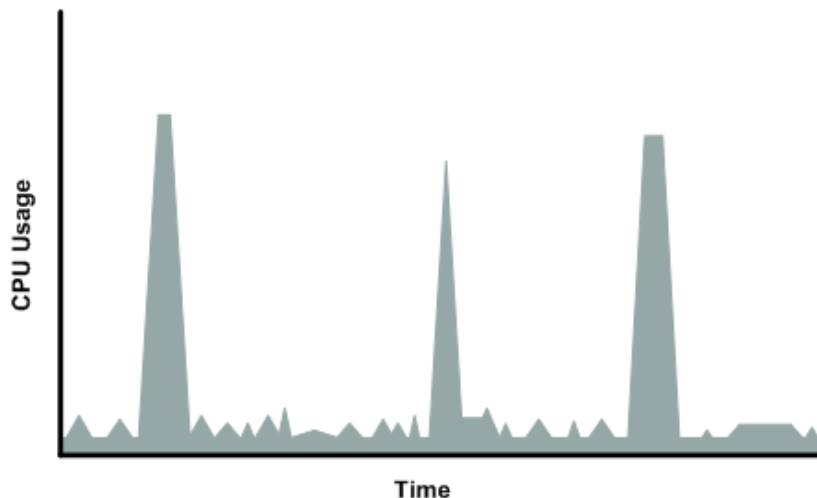
- [Hardware Specifications \(p. 87\)](#)
- [Optimal Application of T1 Micro Instances \(p. 87\)](#)
- [Available CPU Resources During Spikes \(p. 89\)](#)
- [When the Instance Uses Its Allotted Resources \(p. 89\)](#)
- [Comparison with the m1.small Instance Type \(p. 91\)](#)
- [AMI Optimization for Micro Instances \(p. 93\)](#)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Instance Type Details](#).

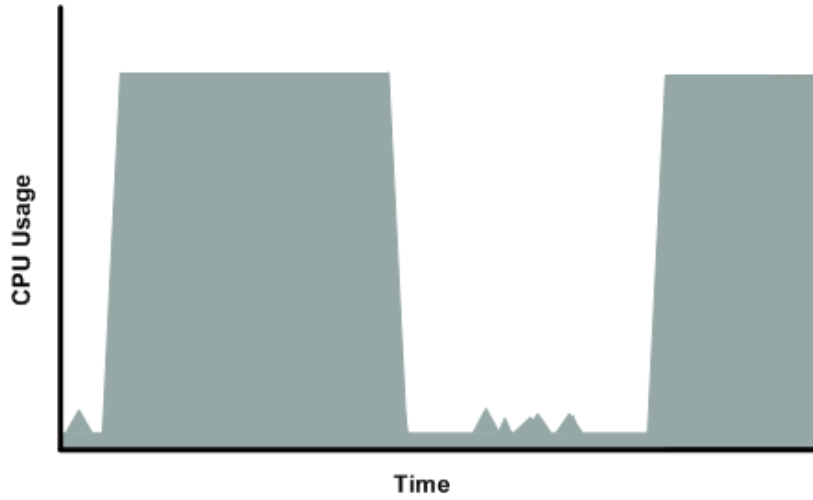
Optimal Application of T1 Micro Instances

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.

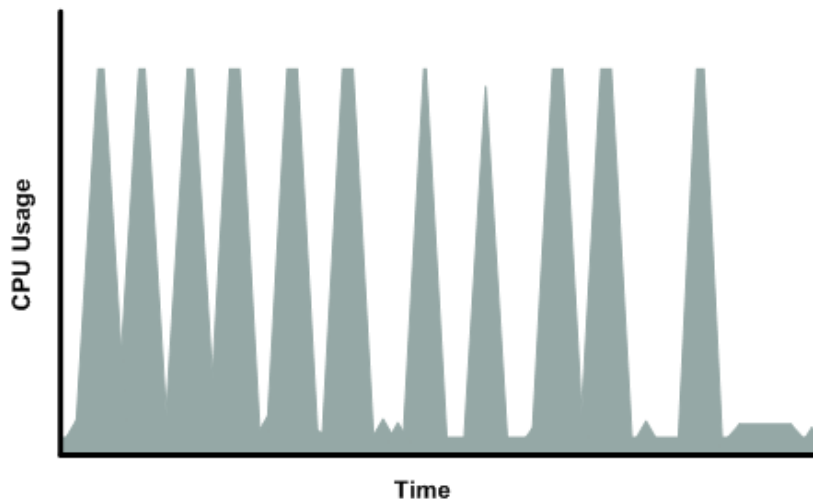


The instance is designed to operate with its CPU usage at essentially only two levels: the normal low background level, and then at brief spiked levels much higher than the background level. We allow the instance to operate at up to 2 EC2 compute units (ECUs) (one ECU provides the equivalent CPU capacity 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.

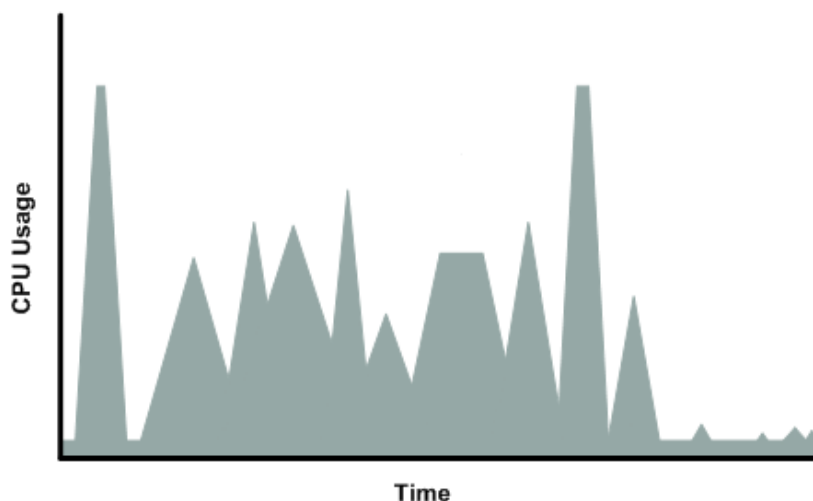
Your application might have a different CPU usage profile than that described in the preceding section. The next figure shows the profile for an application that isn't appropriate for a `t1.micro` instance. The application requires continuous data-crunching CPU resources for each request, resulting in 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. 74).

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.

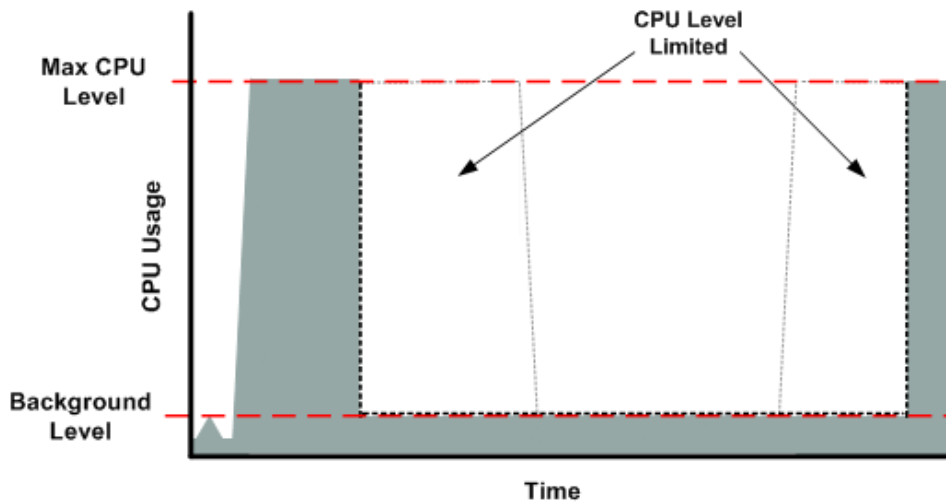
When the Instance Uses Its Allotted Resources

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.

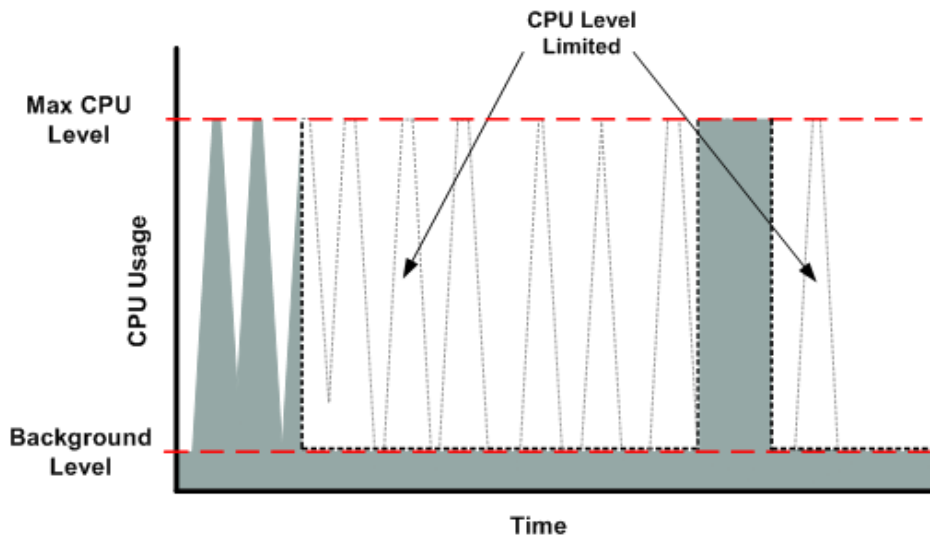
If you enable CloudWatch monitoring for your `t1.micro` instance, you can use the "Avg CPU Utilization" graph in the AWS Management Console to determine whether your instance is regularly using all its allotted CPU resources. We recommend that you look at the maximum value reached during each given period. 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.

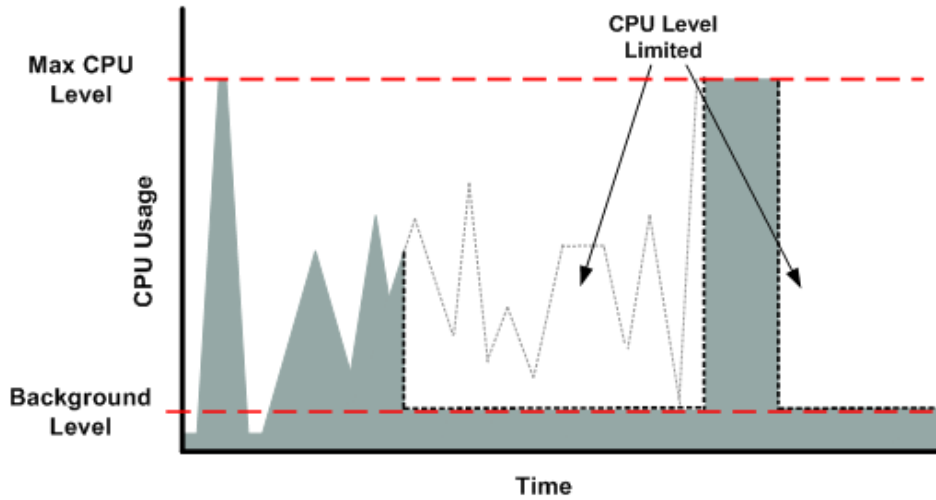
The next figure shows the situation with the long plateaus of data-crunching CPU usage. The CPU hits the maximum allowed level and stays there until the instance's allotted resources are consumed for the period. At that point, we limit the instance to operate at the low background level, and it operates 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.

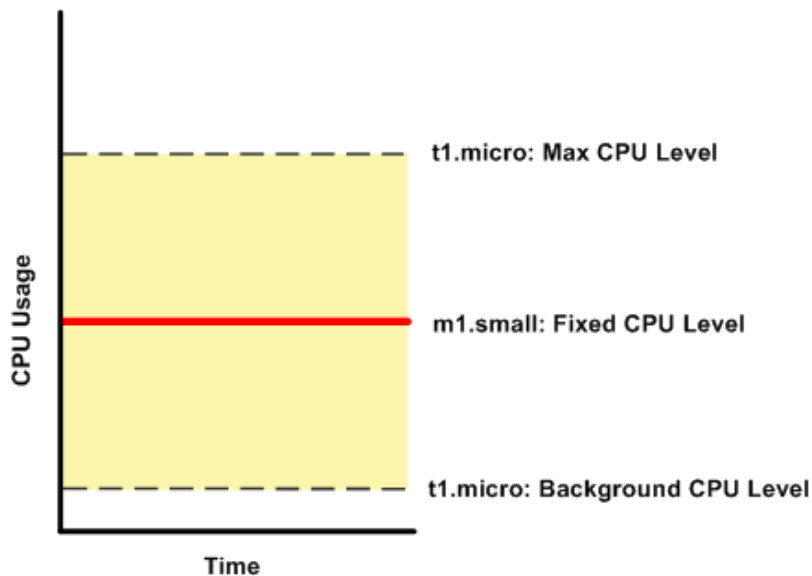


The next figure shows the situation where the background level is too high. Notice that the instance doesn't have to be operating at the maximum CPU level for us to limit it. We limit the instance when it's operating above the normal background level and has consumed its allotted resources for the given 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

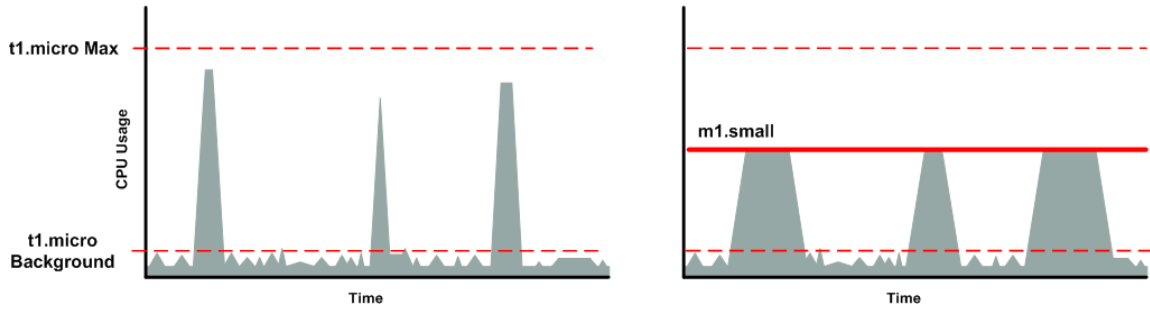
The `t1.micro` instance provides different levels of CPU resources at different times (up to 2 ECUs). By comparison, the `m1.small` instance type provides 1 ECU at all times. The following figure illustrates the difference.



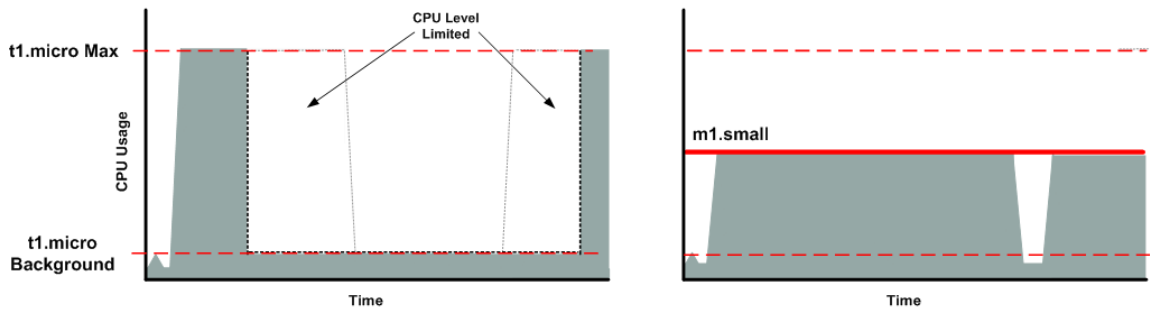
The following figures compare the CPU usage of a `t1.micro` instance with an `m1.small` instance for the various scenarios we've discussed in the preceding sections.

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.

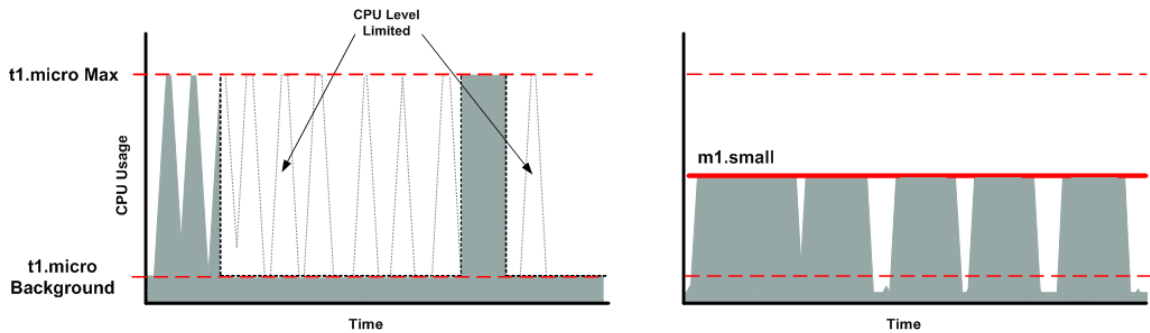
Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
T1 Micro Instances



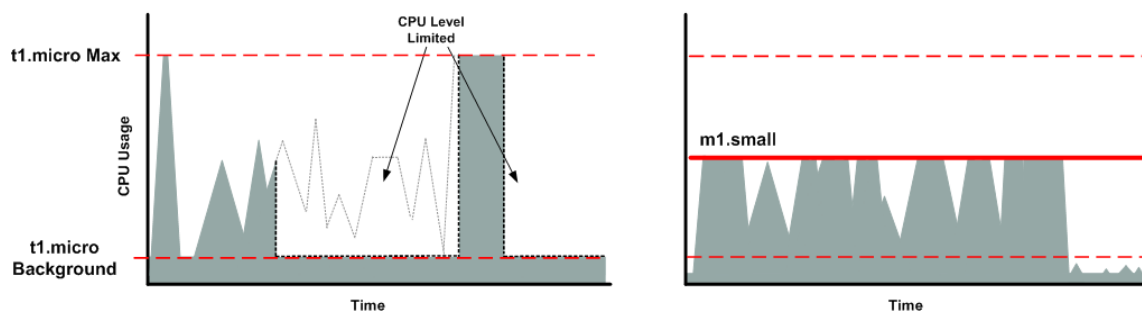
The next figure shows the scenario with the data-crunching requests that used up the allotted resources on the `t1.micro` instance, and how they might look with the `m1.small` instance.



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.

Amazon EBS–Optimized Instances

An Amazon EBS–optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon Elastic Block Store (EBS) I/O. This optimization provides the best performance for your Amazon EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.

When you use an Amazon EBS–optimized instance, you pay an additional low, hourly fee for the dedicated capacity. For more detailed pricing information, see [EBS-optimized Instances](#) on the Amazon EC2 Pricing detail page.

Amazon EBS–optimized instances deliver dedicated throughput to Amazon EBS, with options between 500 Mbps and 2,000 Mbps, depending on the instance type you use. When attached to an Amazon EBS–optimized instance, General Purpose (SSD) volumes are designed to deliver within 10 percent of their of the 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 \(p. 355\)](#).

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 maximum 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 EBS to EC2 connection will become a performance bottleneck.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Placement Groups**

Instance Type	Dedicated EBS 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
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
r3.xlarge	500	4,000	62.5
r3.2xlarge	1,000	8,000	125
r3.4xlarge	2,000	16,000	250

To launch an Amazon EBS–optimized instance, select the **Launch as EBS-optimized instance** option in the launch wizard. If the instance type that you’ve selected can’t be launched as an Amazon EBS–optimized instance, this option is not available.

To launch an Amazon EBS–optimized instance using the AWS CLI, use the [run-instances](#) command with the `--ebs-optimized` option.

To launch an Amazon EBS–optimized instance using Amazon EC2 CLI, use the [ec2-run-instances](#) command with the `--ebs-optimized` option.

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. 346).

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.

Topics

- [Placement Group Limitations \(p. 95\)](#)
- [Launching Instances into a Placement Group \(p. 95\)](#)
- [Deleting a Placement Group \(p. 96\)](#)

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:
 - Compute optimized: `c3.large` | `c3.xlarge` | `c3.2xlarge` | `c3.4xlarge` | `c3.8xlarge` | `cc2.8xlarge`
 - GPU: `cg1.4xlarge` | `g2.2xlarge`
 - Memory optimized: `cr1.8xlarge` | `r3.large` | `r3.xlarge` | `r3.2xlarge` | `r3.4xlarge` | `r3.8xlarge`
 - Storage optimized: `hi1.4xlarge` | `hs1.8xlarge` | `i2.xlarge` | `i2.2xlarge` | `i2.4xlarge` | `i2.8xlarge`
- 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 Virtual Private Cloud 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 an instance into a placement group using the console

1. Open the Amazon EC2 console.
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 \(p. 138\)](#).)
 - 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 \(p. 346\)](#).
 - e. In the navigation pane, click **Instances**, select your instance, click **Actions**, 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 your instances into the placement group.
- a. In the navigation pane, click **Instances**.
 - b. Click **Launch Instance**. Complete the wizard as directed, taking care to select the following:
 - The AMI that you created
 - The number of instances that you'll need
 - The placement group that you created

To create a placement group 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-placement-group](#) (AWS CLI)
- [ec2-create-placement-group](#) (Amazon EC2 CLI)
- [New-EC2PlacementGroup](#) (AWS Tools for Windows PowerShell)

If you prefer, you can use the [ec2-create-image](#) command to create your AMI, the [ec2-create-placement-group](#) command to create your placement group, and use the [ec2-run-instances](#) command to launch an instance into the placement group.

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.
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)

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 `t1.micro` instance is too small for its workload, you can change it to an `m1.small` instance.

The process for resizing an instance varies depends on the type of its root device volume, as follows:

- If the root device for your instance is an Amazon EBS volume, you can easily resize your instance by changing its instance type.
- If the root device for your instance is an instance store volume, you must migrate to a new instance.

To determine the root device type of your instance, open the Amazon EC2 console, click **Instances**, select the instance, and check the value of **Root device type** in the details pane. The value is either `ebs` or `instance store`.

For more information about root device volumes, see [Storage for the Root Device \(p. 48\)](#).

Topics

- [Limitations for Resizing Instances \(p. 97\)](#)
- [Resizing an Amazon EBS-backed Instance \(p. 98\)](#)
- [Resizing an Instance Store-backed Instance \(p. 99\)](#)

Limitations for Resizing Instances

T2 instances must be launched into a VPC using HVM AMIs; they are not supported on the EC2-Classic platform and they do not support PV AMIs. If your account supports EC2-Classic and you have not created any VPCs, you cannot change your instance type to T2 in the console. If your instance uses HVM virtualization and it was launched into a VPC, then you can resize that instance to a T2 instance. For more information, see [T2 Instances \(p. 76\)](#).

All Amazon EC2 instance types support 64-bit AMIs, but only the following instance types support 32-bit AMIs: `t1.micro`, `t2.micro`, `t2.small`, `t1.micro`, `m1.small`, `m1.medium`, and `c1.medium`. If you are resizing a 32-bit instance, you are limited to these instance types.

You cannot add instance store volumes when you resize your instance; instance store volumes may only be added at launch time. If you want to add instance store volumes, consider creating an AMI from your instance and launching a new instance from that AMI with instance store volumes. For more information, see [Amazon EC2 Instance Store](#) (p. 403).

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, we move it to new hardware. If the 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. For more information, see [Stop and Start Your Instance](#) (p. 140).

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, click **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. Click **Actions**, and then click **Stop**.
5. In the confirmation dialog box, click **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, click **Actions**, and then click **Change Instance Type**. Note that this action is disabled if the instance state is not `stopped`.
7. In the **Change Instance Type** dialog box, in the **Instance Type** list, select the type of instance that you need, and then click **Apply**.
8. To restart the stopped instance, select the instance, click **Actions**, and then click **Start**.
9. In the confirmation dialog box, click **Yes, Start**. It can take a few minutes for the instance to enter the `running` state.

[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.

10. [EC2-Classic] If your instance had an associated Elastic IP address, you must reassociate it as follows:
 - a. In the navigation pane, click **Elastic IPs**.
 - b. Select the Elastic IP address that you wrote down before you stopped the instance.
 - c. Click **Associate Address**.
 - d. Select the instance ID that you wrote down before you stopped the instance, and then click **Associate**.

Resizing an Instance Store-backed Instance

You can create an image from your current instance, launch a new instance from this image with the instance type you need, and then terminate the original instance that you no longer 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 current instance and associate it with the new instance.

To resize an instance store-backed instance

1. (Optional) If the instance you are resizing has an associated Elastic IP address, record the Elastic IP address now so that you can associate it with the resized instance later.
2. 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 \(p. 64\)](#). When you are finished creating a new AMI from your instance, return to this procedure.
3. Open the Amazon EC2 console and in the navigation pane, select **AMIs**. From the filter lists, select **Owned by me**, and select the image 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 the console displays the region that you created your AMI in.

4. Click **Launch**. When you specify options in the launch wizard, be sure to specify the new instance type that you need. It can take a few minutes for the instance to enter the `running` state.
5. (Optional) 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, click **Elastic IPs**.
 - b. Select the Elastic IP address that you recorded at the beginning of this procedure.
 - c. Click **Associate Address**.
 - d. Select the instance ID of the new instance, and then click **Associate**.
6. (Optional) You can terminate the instance that you started with, if it's no longer needed. Select the instance and check its instance ID against the instance ID that you wrote down at the beginning of this procedure to verify that you are terminating the correct instance. Click **Actions**, and then click **Terminate**.

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. 103).

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. 107).

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.

Because you can access instance metadata and user data from within your running instance, you do not need to use the Amazon EC2 console or the CLI tools. This can be helpful when you're writing scripts to run from within your instance. For example, you can access your instance's local IP address from within the running instance to manage a connection to an external application.

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.

For more information about adding user data when you launch an instance, see [Launching an Instance](#) (p. 130). You can add or modify user data on Amazon EBS-backed instances when they're stopped. For more information about adding user data to a stopped instance, see [Modifying a Stopped Instance](#) (p. 142).

When you are adding user data, take note of 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 API command line tools perform the base64 encoding for you. The data is decoded before being presented to the instance. For more information about base64 encodings, go to <http://tools.ietf.org/html/rfc4648>.

Topics

- [Retrieving Instance Metadata](#) (p. 100)
- [Retrieving User Data](#) (p. 102)
- [Retrieving Dynamic Data](#) (p. 103)
- [Instance Metadata Categories](#) (p. 103)

Retrieving Instance Metadata

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 \(p. 103\)](#).

```
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/inter  
faces/macs/02:29:96:8f:6a:2d/subnet-id  
subnet-be9b61d7
```

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

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Instance Metadata Categories**

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 <code>ancestor-amis</code> 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 <i>N</i>	The virtual devices associated with Amazon EBS volumes, if any are present. This value is only available in metadata if it is present at launch time. The <i>N</i> indicates the index of the Amazon EBS volume (such as <code>ebs1</code> or <code>ebs2</code>).	2007-12-15
block-device-mapping/ephemeral <i>N</i>	The virtual devices associated with ephemeral devices, if any are present. The <i>N</i> 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 <code>swap</code> . 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 <code>eth0</code> device (the device for which the device number is 0).	1.0
iam/info	Returns information about the last time the instance profile was updated, including the instance's <code>LastUpdated</code> date, <code>InstanceProfileArn</code> , and <code>InstanceProfileId</code> .	2012-01-12
iam/security-credentials /role-name	Where <i>role-name</i> is the name of the IAM role associated with the instance. Returns the temporary security credentials (<code>AccessKeyId</code> , <code>SecretAccessKey</code> , <code>SessionToken</code> , and <code>Expiration</code>) associated with the IAM role.	2012-01-12
instance-action	Notifies the instance that it should reboot in preparation for bundling. Valid values: <code>none</code> <code>shutdown</code> <code>bundle-pending</code> .	2008-09-01
instance-id	The ID of this instance.	1.0

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Instance Metadata Categories**

Data	Description	Version Introduced
instance-type	The type of instance. For more information, see Instance Types (p. 74) .	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 device number associated with that interface. Each interface must have a unique device number. The device number serves as a hint to device naming in the instance; for example, device-number is 2 for the eth2 device.	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
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

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Instance Metadata Categories**

Data	Description	Version Introduced
network/interfaces/mac/mac/public-hostname	The interface's public DNS. If the instance is in a VPC, this category is only returned if the <code>enableDnsHostnames</code> attribute is set to <code>true</code> . For more information, see Using DNS with Your VPC .	2011-01-01
network/interfaces/mac/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/mac/mac/security-groups	Security groups to which the network interface belongs. Returned only for instances launched into a VPC.	2011-01-01
network/interfaces/mac/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/mac/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/mac/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/mac/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/mac/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 <code>enableDnsHostnames</code> attribute is set to <code>true</code> . For more information, see Using DNS with Your VPC .	2007-01-19
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

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Importing and Exporting Instances**

Data	Description	Version Introduced
ramdisk-id	The ID of the RAM disk specified at launch time, if applicable.	2007-10-10
reservation-id	ID of the reservation.	1.0
security-groups	The names of the security groups applied to the instance. Note Only instances launched into a VPC can change security groups after launch. These changes will be reflected here and in network/interfaces/security-groups .	1.0
services/domain	The domain for AWS resources for the region; for example, <code>amazonaws.com</code> for <code>us-east-1</code> .	2014-02-25

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: <code>enabled</code> <code>disabled</code>	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

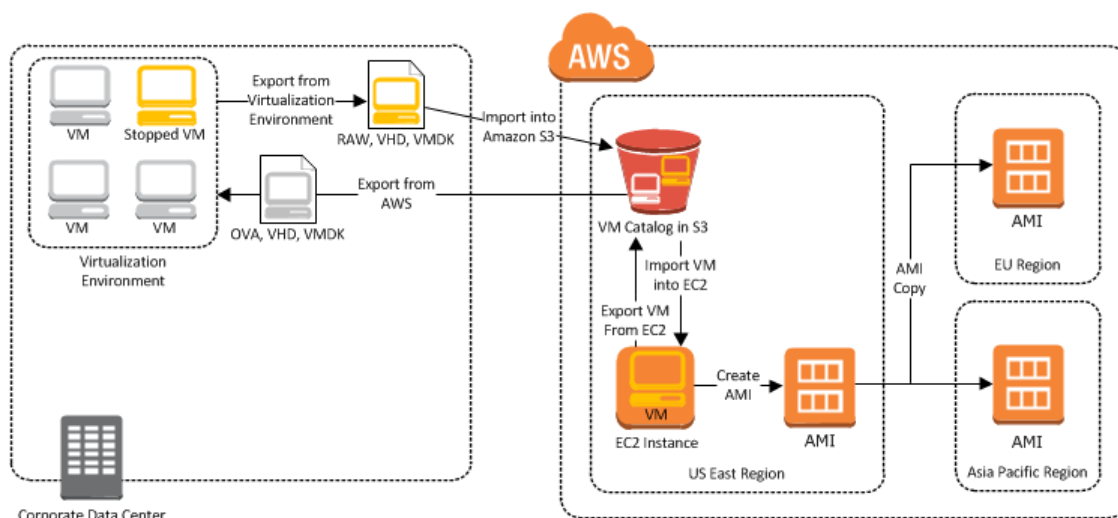
Importing and Exporting Instances

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 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 virtual machines 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 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*.

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. After you import your applications and workloads into an Amazon EC2 instance, you can create Amazon Machine Images (AMIs) and run multiple copies of the same image. You can also create snapshots and use them to back up your data. You can use AMI and snapshot copies to replicate your applications and workloads around the world. For more information about AMI copy, see [Copying an AMI \(p. 67\)](#).
- **Copy your VM image catalog to Amazon EC2**—You can copy 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 Amazon EC2 instances 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 resulting Amazon EC2 instances to create Amazon EC2 AMIs. You can use the AMIs as your image catalog within Amazon EC2.
- **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 store the imported images as Amazon Elastic Block Store (Amazon EBS)-backed AMIs so they're 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.

The following diagram shows the process of exporting a VM from your on-premises virtualization environment to AWS.



Contents

- [VM Import/Export Prerequisites \(p. 108\)](#)
- [Importing a VM into Amazon EC2 \(p. 111\)](#)
- [Exporting Amazon EC2 Instances \(p. 120\)](#)
- [Troubleshooting VM Import/Export \(p. 121\)](#)

VM Import/Export Prerequisites

Before you begin the process of exporting an instance 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.

If you plan to use the command line tools to export your instance, you must also download and install them. For more information, see [Setting Up the Amazon EC2 Tools](#).

Contents

- [Operating Systems \(p. 109\)](#)
- [Image Formats \(p. 109\)](#)
- [Instance Types \(p. 110\)](#)
- [Requirements and Limitations \(p. 110\)](#)

Operating Systems

The following operating systems can be imported into and exported from Amazon EC2.

Windows (32- and 64-bit)

- Microsoft Windows Server 2012 R2 (Standard)
- Microsoft Windows Server 2012 (Standard, Datacenter)
- Microsoft Windows Server 2008 R2 (Standard, Datacenter, Enterprise)
- Microsoft Windows Server 2008 (Standard, Datacenter, Enterprise)
- Microsoft Windows Server 2003 R2 (Standard, Datacenter, Enterprise)
- Microsoft Windows Server 2003 (Standard, Datacenter, Enterprise) with Service Pack 1 (SP1) or later

Linux/Unix (64-bit)

- Red Hat Enterprise Linux (RHEL) 5.1-5.10, 6.1-6.5

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 5.1-5.10, 6.1-6.5

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
- 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 volumes and instances into Amazon EC2:

- RAW format for importing volumes and instances.
- Virtual Hard Disk (VHD) image formats, which are compatible with Microsoft Hyper-V and Citrix Xen virtualization products.
- ESX Virtual Machine Disk (VMDK) image formats, which are 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.

Exporting Image Formats from Amazon EC2

AWS supports the following image formats for exporting both volumes and instances from Amazon EC2:

- 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`

Requirements and Limitations

Known Limitations for Importing a VM into Amazon EC2

Importing instances and volumes is subject to the following limitations:

- You can have up to five import tasks in progress at the same time per region.
- 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 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.

- 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.

- Imported instances automatically have access to the Amazon EC2 instance store, which is temporary disk storage from 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 \(p. 403\)](#).
- Tasks must complete within 7 days of the start date.
- 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 \(p. 346\)](#). For Microsoft Windows Server 2012 R2 VMs, SR-IOV driver are automatically installed as a part of the import process.

Known Limitations for Exporting a VM from Amazon EC2

Exporting instances and volumes is subject to the following limitations:

- You cannot export Amazon Elastic Block Store (Amazon EBS) data volumes.
- You cannot export an instance that has more than one virtual disk.
- You cannot export an instance that has more than one network interface.
- You cannot export an instance from Amazon EC2 unless you previously imported it into Amazon EC2 from another virtualization environment.

Importing a VM into Amazon EC2

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. For more information about how to launch an Amazon EC2 instance from an AMI, see [Launch Your Instance \(p. 129\)](#).

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*.

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 \(p. 112\)](#).
2. Prepare the VM for import to Amazon EC2. For more information, see [Step 2: Prepare Your VM \(p. 112\)](#).
3. Export the VM from the virtualization environment. For more information, see [Step 3: Export Your VM from Its Virtual Environment \(p. 113\)](#).
4. Import the VM into Amazon EC2. For information, see [Step 4: Importing Your VM into Amazon EC2 \(p. 113\)](#).

5. Launch the instance in Amazon EC2. For more information, see [Step 5: Launch the instance in Amazon EC2 \(p. 119\)](#).

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 Elastic Compute Cloud 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 \(p. 108\)](#).
- 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 installed, as required by [Amazon Windows EC2Config Service](#).
- Do not run System Preparation (Sysprep) on your Windows VM images. We recommend that you import the image and then use the Amazon EC2 Config service to run Sysprep.
- 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.

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 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.
- 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.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Importing a VM into Amazon EC2**

- 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 [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 architecture -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. 118).

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/bPxrFiCYEXAMPLEKEY -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.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Importing a VM into Amazon EC2**

```
$ 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 Elastic Compute Cloud 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.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Importing a VM into Amazon EC2**

```
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 Elastic Compute Cloud 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 \(p. 264\)](#).

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Importing a VM into Amazon EC2**

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 availability_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/bPxRfiCYEXAMPLEKEY --region us-east-1 -o AKIAI44QH8DHBEXAMPLE -w je7MtGbClwBF/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
```

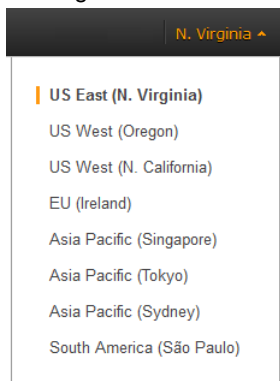
Step 5: Launch the instance in Amazon EC2

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. 126).

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, under **INSTANCES**, click **Instances**.
4. In the content pane, right-click the instance, 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.

Contents

- [Export an Instance \(p. 120\)](#)
- [Cancel or Stop the Export of an Instance \(p. 121\)](#)

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*.
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 \(p. 108\)](#) 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. \(p. 122\)](#)
- [Client.UnsupportedOperation: This instance has multiple volumes attached. Please remove additional volumes. \(p. 122\)](#)
- [ClientError: Footers not identical \(p. 122\)](#)
- [ClientError: Uncompressed data has invalid length. \(p. 122\)](#)
- [ERROR: Bucket <MyBucketName> is not in the <RegionName> region, it's in <RegionName>. \(p. 122\)](#)
- [ERROR: File uses unsupported compression algorithm 0. \(p. 123\)](#)
- [Error starting instances: Invalid value <instance ID> for instancedId. Instance does not have a volume attached at root \(/dev/sda1\). \(p. 123\)](#)

- [java.lang.OutOfMemoryError: Java heap space](#) (p. 123)
- [Service.InternalError: An internal error has occurred. Status Code: 500, AWS Service: AmazonEC2](#) (p. 123)
- [FirstBootFailure: This import request failed because the Windows instance failed to boot and establish network connectivity.](#) (p. 123)
- [Linux is not supported on the requested instance](#) (p. 125)

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.

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.

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

- [The installation of Windows is not valid on the virtual machine \(p. 124\)](#)
- [TCP/IP networking and DHCP are not enabled \(p. 124\)](#)
- [A volume that Windows requires is missing from the virtual machine \(p. 124\)](#)
- [Windows always boots into System Recovery Options \(p. 125\)](#)
- [The virtual machine was created using a physical-to-virtual \(P2V\) conversion process \(p. 125\)](#)
- [Windows activation fails \(p. 125\)](#)
- [No bootable partition found \(p. 125\)](#)

The installation of Windows is not valid on the virtual machine

Cause: The installation of Windows must be valid before you can successfully import the virtual machine.

Resolution: Do not run System Preparation (Sysprep) before shutting down the EC2 instance. After the instance is imported, you can run Sysprep from the instance before you create an AMI. Importing creates a single instance, so running Sysprep is not necessary.

Ensure that the installation process is fully complete and that Windows boots (without user intervention) to a login prompt.

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 be 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.

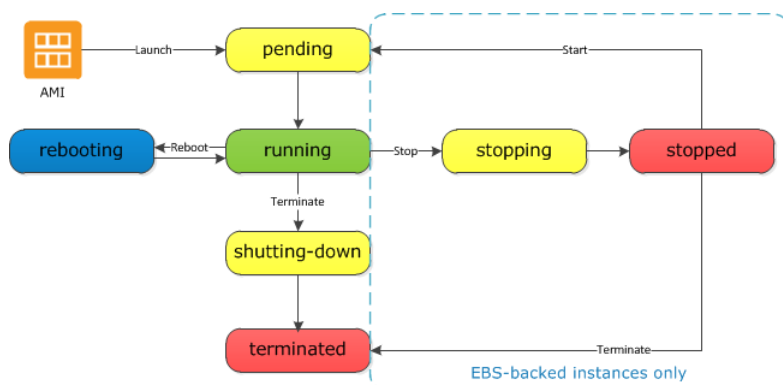
Resolution: Retry using one of the supported instance types.

- 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 \(p. 48\)](#).



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 starts to boot, 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 \(p. 129\)](#) and [Connecting to Your Windows Instance Using RDP \(p. 138\)](#).

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 \(p. 140\)](#).

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 \(p. 143\)](#).

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 \(p. 143\)](#).

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 preserve volumes that you attach to a running instance and delete volumes that you attach at launch, such as the root volume.

For more information, see [Terminate Your Instance](#) (p. 145).

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

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Terminate
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
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 <code>stopping</code> . Each time an instance transitions from <code>stopped</code> to <code>pending</code> , we start a new instance billing hour.	You stop incurring charges for an instance as soon as its state changes to <code>shutting-down</code> .

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 \(p. 148\)](#).

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 Usage Tier](#). You can either leverage the Free Usage Tier to launch and use a micro instance for free for 12 months. If you launch an instance that is not within the Free Usage 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. 130)
Use the Amazon EC2 console with an Amazon EBS snapshot that you created	Launching an Instance from a Backup (p. 136)
Use the Amazon EC2 console with an AMI that you purchased from the AWS Marketplace	Launching an AWS Marketplace Instance (p. 136)
Use the AWS CLI with an AMI that you select	Using Amazon EC2 through the AWS CLI

Method	Documentation
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 \(p. 138\)](#).

When you are finished with an instance, be sure to terminate it. For more information, see [Terminate Your Instance \(p. 145\)](#).

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 \(p. 312\)](#). 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 Usage Tier](#), you are charged for the time that the instance is running, even if it remains idle.

Topics

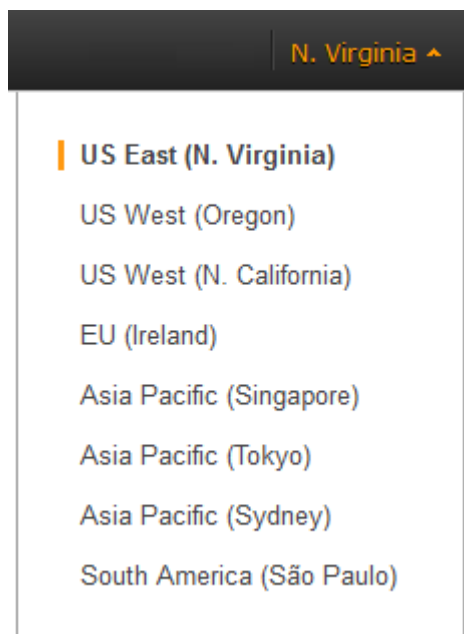
- [Launching Your Instance from an AMI \(p. 130\)](#)
- [Launching Instances Using an Existing Instance as a Template \(p. 135\)](#)

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 \(p. 424\)](#).



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 \(p. 136\)](#).

Community AMIs

The AMIs that AWS community members have made available for others to use. To filter the list of AMIs by operating system, select the **Windows** 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 \(p. 48\)](#).
 - 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 \(p. 74\)](#).

To remain eligible for the free tier, select the **t2.micro** instance type. For more information, see [T2 Instances \(p. 76\)](#).

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.
- 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 \(p. 312\)](#). 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 \(p. 321\)](#).
- **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 \(p. 302\)](#).
- **Shutdown behavior:** Select whether the instance should stop or terminate when shut down. For more information, see [Changing the Instance Initiated Shutdown Behavior \(p. 148\)](#).
- **Enable termination protection:** Select this check box to prevent accidental termination. For more information, see [Enabling Termination Protection for an Instance \(p. 147\)](#).
- **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 \(p. 197\)](#).
- **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 \(p. 93\)](#).

- **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 Virtual Private Cloud 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 \(ENI\) \(p. 334\)](#). 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 \(p. 324\)](#).
 - **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 \(p. 94\)](#).
 - **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 \(p. 403\)](#) and [Amazon EBS Volumes \(p. 353\)](#).
 - **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 usage tier, you need to keep under 30 GiB of total storage to stay within the free usage tier.

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 a Volume \(p. 377\)](#).

- **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 \(p. 355\)](#).

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 \(p. 355\)](#).

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 SSD volumes such as Provisioned IOPS (SSD) and General Purpose (SSD). If you are unable to create an SSD volume (or

launch an instance with an 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 General Purpose (SSD) and Provisioned IOPS (SSD) volumes by creating a 1 GiB General Purpose (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 \(p. 148\)](#).
- **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 \(p. 387\)](#).

Note

Encrypted boot volumes are not supported at this time.

8. On the **Tag Instance** page, specify [tags \(p. 427\)](#) 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 \(p. 264\)](#).) 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 groups 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 you to connect to your instance over RDP (port 3389).

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

We recommend against selecting the **Proceed without key pair** option. If you launch an instance without a key pair, you won't be able to connect to it. This option is used only when you are creating your own AMI and don't need to connect to the instance.

12. 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. 488).

Launching Instances 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
- Tags associated with the instance, if applicable
- Kernel ID and RAM disk ID, if applicable
- IAM role associated with the instance, if applicable
- Security group associated with the instance
- Tenancy setting, if launching into a VPC (shared or dedicated)
- Amazon EBS-optimization setting (true or false)

The following configuration details are not copied from your selected instance; instead, the wizard applies their default settings or behavior:

- Storage: The default storage configuration is determined by the AMI and the instance type.
- Public IP address: The option to assign a public IP address to your instance is enabled by default when launching into a default subnet.
- Termination protection: Disabled by default.
- Shutdown behavior: Set to 'stop' by default.
- User data: None by default.

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 Instance from a Backup

At this time, although you can create a Windows AMI from a snapshot, you can't launch an instance from the AMI.

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 \(p. 58\)](#). 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 \(p. 61\)](#).

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 \(p. 130\)](#). 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 RDP (port 3389). We recommend that you adjust these rules to allow only a specific IP address or range of addresses to access your instance over those specific 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 \(p. 138\)](#).

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 \(p. 61\)](#).

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 .
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. 487).

The following instructions explain how to connect to your instance using an RDP client.

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 [Microsoft's Remote Desktop Connection Client](#), or the Microsoft Remote Desktop app from the Apple iTunes store. For Linux, you can use [rdesktop](#).

- **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 Instances](#) (p. 308).

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

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](#).

Important

If you are connecting to a Windows 2012 R2 instance using Mac OS X, the Remote Desktop Connection client from the Microsoft website may not work. Use the Microsoft Remote Desktop app from the Apple iTunes store instead.

To connect to your Windows instance

1. In the Amazon EC2 console, select the instance, and then click **Connect**.
2. In the **Connect To Your Instance** dialog box, click **Get Password** (it will take a few minutes after the instance is launched before the password is available).
3. Click **Browse** and navigate to the private key file you created when you launched the instance. Select the file and click **Open** to copy the entire contents of the file into contents box.
4. Click **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. Click **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 click **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 double-click 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, click **Connect** to connect to your instance. If you are using **Microsoft Remote Desktop** on a Mac, skip the next step.
8. Log in to the instance as prompted, using the default **Administrator** account and the default administrator password that you recorded or copied previously. If your **Remote Desktop Connection** already has an Administrator account set up, you may need to click 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 click **Yes** or **Continue** to continue if you trust the certificate.
 - a. If you are using **Remote Desktop Connection** from a Windows PC, click **View certificate**. If you are using **Microsoft Remote Desktop** on a Mac, click **Show Certificate**.
 - b. Click 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, click **Actions**, and then click **Get System Log**.
 - d. In the system log output, look for an entry labelled `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 click **OK**. If you are using **Microsoft Remote Desktop** on a Mac, return to the **Verify Certificate** and click **Continue**.
 - f. If you are using **Remote Desktop Connection** from a Windows PC, click **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 from Windows

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 Windows computer using the local file sharing feature of Windows Remote Desktop. If you enable this option in your Windows Remote Desktop Connection software, 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 information about this feature, go to the [Microsoft Support website](#) or [The most useful feature of Remote Desktop I never knew about](#) on the MSDN Blogs website.

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. 145).

Topics

- [Overview](#) (p. 140)
- [Stopping and Starting Your Instances](#) (p. 141)
- [Modifying a Stopped Instance](#) (p. 142)

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. 49).

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 (EIP) 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 (EIP). 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 \(p. 151\)](#).
- 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-Registering and Registering Amazon EC2 Instances](#) in the *Elastic Load Balancing Developer Guide*.

For more information, see [Differences Between Reboot, Stop, and Terminate \(p. 128\)](#).

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 \(p. 148\)](#).

To stop and start an Amazon EBS-backed instance using the console

1. In the navigation pane, click **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. Click **Actions**, and then click **Stop**. If **Stop** is disabled, either the instance is already stopped or its root device is an instance store volume.

4. In the confirmation dialog box, click **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 \(p. 142\)](#).
6. To restart the stopped instance, select the instance, click **Actions**, and then click **Start**.
7. In the confirmation dialog box, click **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, click **Elastic IPs**.
 - b. Select the Elastic IP address that you wrote down before you stopped the instance.
 - c. Click **Associate Address**.
 - d. Select the instance ID that you wrote down before you stopped the instance, and then click **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 and user data attributes 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

1. In the navigation pane, click **Instances**.
2. Select the stopped instance, click **Actions**, and then click **Change Instance Type**.
3. In the **Change Instance Type** dialog box, in the **Instance Type** list, select the type of instance you need, and then click **Apply**.

For more information, see [Resizing Your Instance \(p. 97\)](#).

To change the user data for a stopped instance using the console

1. In the navigation pane, click **Instances**.
2. Select the stopped instance, click **Actions**, and then click **View/Change User Data**.
3. In the **View/Change User Data** dialog box, update the user data, and then click **Save**. Note that you can't change the user data if the instance is running, but you can view it.

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)

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 [Monitoring Events for Your Instances \(p. 194\)](#).

We recommend that you use Amazon EC2 to reboot your instance instead of running the operating system reboot command from your instance.

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**, 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 \(p. 144\)](#)
- [Working with Instances Scheduled for Retirement \(p. 144\)](#)

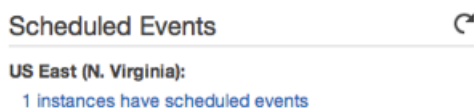
For more information about types of instance events, see [Monitoring Events for Your Instances \(p. 194\)](#).

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 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 \(p. 144\)](#).
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. 140) .
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. 62) .
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. 64) .

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 \(p. 140\)](#). For more information, see [Differences Between Reboot, Stop, and Terminate \(p. 128\)](#).

Topics

- [Instance Termination](#) (p. 146)
- [Terminating an Instance](#) (p. 146)
- [Enabling Termination Protection for an Instance](#) (p. 147)
- [Changing the Instance Initiated Shutdown Behavior](#) (p. 148)
- [Preserving Amazon EBS Volumes on Instance Termination](#) (p. 148)

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, any Amazon EBS volumes that you attach as you launch the instance are automatically deleted when the instance terminates. However, by default, any volumes that you attach to a running 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. 148).

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. 147).

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. 148).

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**, 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. 148).

Instances that are part of an Auto Scaling group are not covered by termination protection. For more information, see [Instance Termination Policy for Your Auto Scaling Group](#) in the *Auto Scaling Developer Guide*.

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**, 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)

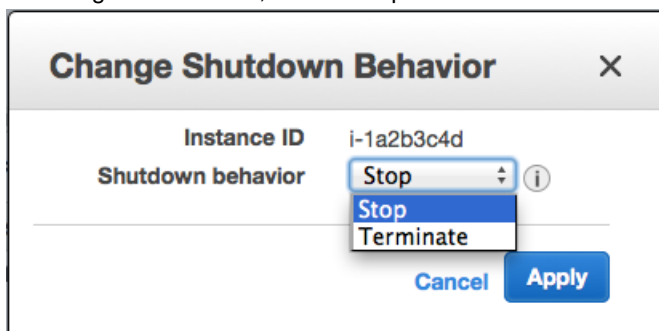
Changing the Instance Initiated Shutdown Behavior

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**, 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 volumes that you attach to a running instance even after the instance terminates
- Preserve any volumes that you attach to your instance at launch when you stop and restart an instance
- Delete the volumes that you attach to your instance at launch, including the root device volume, when you terminate the instance

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Preserving Amazon EBS Volumes on Instance Termination**

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; otherwise, we preserve the volume. If the `DeleteOnTermination` attribute of a volume 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.

If you detach a volume that you attached to your instance at launch, and then reattach it, we preserve it even after the instance terminates. In other words, 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 Amazon EBS Volumes in an Instance Block Device Mapping \(p. 419\)](#).

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}}]"
```

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Preserving Amazon EBS Volumes on Instance Termina-
tion**

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
```

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 \(p. 151\)](#)
- [Upgrading PV Drivers on Your Windows AMI \(p. 164\)](#)
- [Setting Passwords for Windows Instances \(p. 174\)](#)
- [Configuring a Secondary Private IP Address for Your Windows Instance in a VPC \(p. 179\)](#)
- [Setting the Time for a Windows Instance \(p. 183\)](#)

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 CloudWatch Logs, see [Monitoring System, Application, and Custom Log Files](#) in the Amazon CloudWatch Developer Guide. The service binaries and additional files are contained in the %Program-Files%\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 about Sysprep, see [Sysprep Technical Reference](#).

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 \(p. 62\)](#).

Contents

- [Overview of EC2Config Tasks \(p. 152\)](#)
- [Ec2 Service Properties \(p. 153\)](#)
- [EC2Config Settings Files \(p. 159\)](#)
- [Installing the Latest Version of EC2Config \(p. 162\)](#)
- [Stopping, Deleting, or Uninstalling EC2Config \(p. 163\)](#)
- [Troubleshooting CloudWatch Logs in EC2Config \(p. 164\)](#)

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:

- Set the computer host name to match the private DNS name (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 any 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 [Creating an Amazon EBS-Backed Windows AMI \(p. 62\)](#).

EC2Config creates a WMI object that you can use to detect when Windows is ready. You can get the value of `ConfigurationComplete` as follows, and test whether it is `true`.

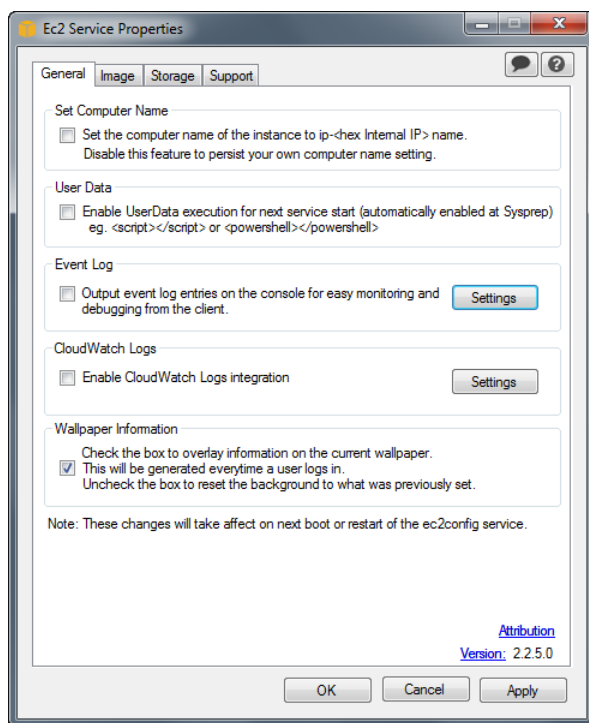
```
(Get-WmiObject -Namespace root\Amazon -Class EC2_ConfigService).ConfigurationComplete
```

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 EC2Config to execute user data, you must enclose the lines of the script within one of the following special tags:

```
<script></script>
```

Run any command that you can run at the `cmd.exe` prompt.

Example: `<script>dir > c:\test.log</script>`

```
<powershell></powershell>
```

Run any command that you can run at the Windows PowerShell 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 my-Folder/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.

```
$UserData = [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($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 `Settings.xml` using a script similar to the following:

```
<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 "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>
```

Event Log

Use this setting to display event log entries on the console during boot for easy monitoring and debugging.

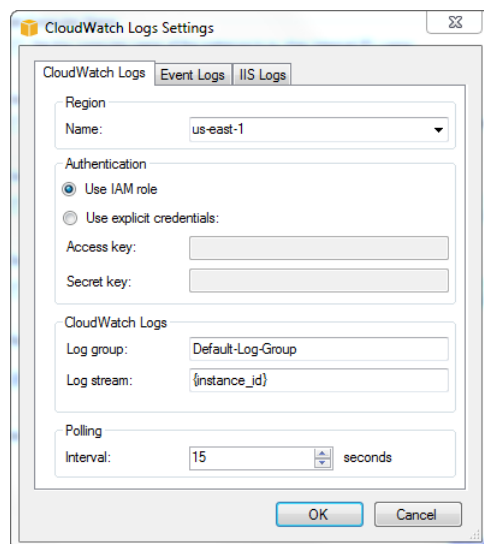
Click **Settings** to specify filters for the log entries sent to the console. By default, the three most recent error entries from the system event log are sent to the console.

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. 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 **Settings**.

Amazon Elastic Compute Cloud User Guide for Microsoft Windows Ec2 Service Properties



2. In the **CloudWatch Settings** dialog box, under **Region**, set the region to **us-east-1**.

Note

CloudWatch Logs is only supported in the US East (Northern Virginia) Region.

3. Under **Authentication**, supply your AWS credentials (leave the **Access key** and **Secret key** fields blank if you have an IAM role configured for this instance and will use that credential to upload logs).
4. Under **CloudWatch Logs**, in the **Log group** field, type the log group name with which to associate these logs. If you enter a log group name that doesn't already exist, CloudWatch Logs automatically creates it for you.
5. In the **Log stream** field, type a name for the 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.
6. Under **Polling**, change the interval if appropriate. To prevent excessive CPU and IO operations on Windows Server 2003 instances, set this to 300 seconds (5 minutes).
7. Click the **Event Logs** tab, and then choose the type of log messages to send to CloudWatch Logs.
8. Click the **IIS Logs** tab, and then under **Log Directories**, select the folder where IIS logs are stored for an individual site (e.g., `C:\inetpub\logs\LogFiles\W3SVCn`). To browse for folders not in the list, click **Add**, and then select the folder.
9. Click **OK** to save your settings.

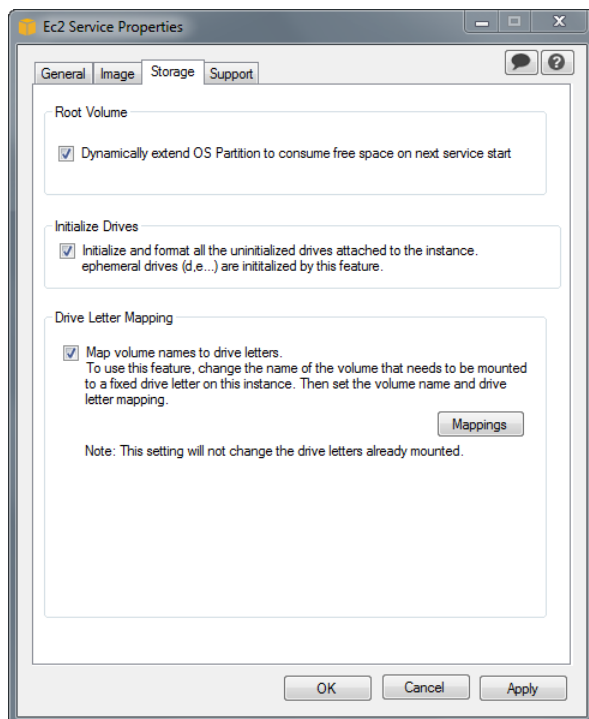
Wallpaper Information

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 instance store 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:.

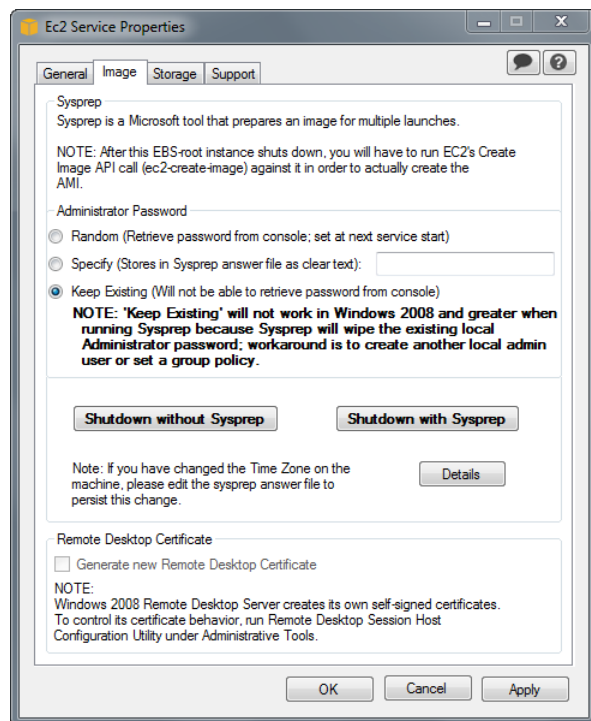
To choose the drive letters for your volumes, click **Mappings**. In the **DriveLetterSetting** dialog box, specify the **Volume Name** and **Drive Letter** values for each volume, and then click **OK**. We recommend that you select drive letters that avoid conflicts with drive letters that are likely 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.

Amazon Elastic Compute Cloud User Guide for Microsoft Windows Ec2 Service Properties

Otherwise, if you have finished customizing your instance and are ready to create your AMI from this instance, click the **Image** tab.



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.

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 \(p. 159\)](#). For an example of configuring

and then running Sysprep from the command line, see `Ec2ConfigService\Scripts\InstallUpdates.ps1`.

EC2Config Settings Files

The settings files control the operation of the EC2Config service. These files are located in the `Ec2ConfigService\Settings` directory:

- `ActivationSettings.xml`—Controls product activation using a key management server (KMS).
- `BundleConfig.xml`—Controls how EC2Config prepares an instance for AMI creation.
- `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 instance store volumes during startup. This feature is enabled by default and initializes and mounts the instance store volumes as drives D:, E:, and so on. For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 403\)](#).
- `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`—Indicates whether to search through the DNS Suffix List for appropriate KMS entries. When the appropriate KMS entries are found, the plug-in sets your activation server to the first server to respond to the request successfully. Starting with Windows Server 2008 R2, Windows Server is able to search the suffix list automatically. Otherwise, the plug-in performs this search manually.

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>
```

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, 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**.

If you don't need to update the configuration settings or create your own AMI, you can delete the service. Deleting a service removes its registry subkey.

To delete the EC2Config service

1. Start a command prompt window.
2. Run the following command:

```
C:\> sc delete ec2config
```

If you don't need to update the configuration settings or create your own AMI, you can uninstall EC2Config. Uninstalling a service removes the files, the registry subkey, and any shortcuts to the service.

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**.

Troubleshooting CloudWatch Logs in EC2Config

I cannot see logs in the Amazon 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 Amazon CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
3. On the navigation bar, select the **US East (N. Virginia)** 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 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 -RetentionInDays 3653
```

The Enable CloudWatch Logs integration checkbox 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 version 2.2.6 or later.

Upgrading PV Drivers on Your Windows AMI

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.

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

- [Xen Drivers \(p. 165\)](#)
- [Upgrading PV Drivers on Your Windows Server 2008 and 2008 R2 Instances \(p. 167\)](#)
- [Upgrading Your Citrix Xen Guest Agent Service \(p. 169\)](#)
- [Upgrading PV Drivers on Your Windows Server 2003 Instance \(p. 170\)](#)
- [Troubleshooting \(p. 171\)](#)

Xen Drivers

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. The particular Xen driver on your instance depends on when its AMI was created.

The following table shows key differences between the different drivers.

Characteristic	RedHat PV	AWS 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.
Attached volumes	Supports up to 16 attached volumes.	Supports up to 16 attached volumes.

Characteristic	RedHat PV	AWS V P
Network	The driver has known issues where the network connection resets under high loads; for example, fast FTP file transfers.	The driver has known issues where the network connection resets under high loads; for example, fast FTP file transfers.

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).ToString("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, xenkbd, and xenvid.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Upgrading PV Drivers on Your Windows Server 2008
and 2008 R2 Instances**

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.

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 \(p. 172\)](#).

RedHat 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.

For more information about upgrading your RedHat drivers on an existing AMI to Citrix drivers, see [Upgrading PV Drivers on Your Windows AMI \(p. 164\)](#).

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 \(p. 151\)](#).
- 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 \(p. 62\)](#). 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 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 \(p. 138\)](#).
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.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Upgrading PV Drivers on Your Windows Server 2008
and 2008 R2 Instances**

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

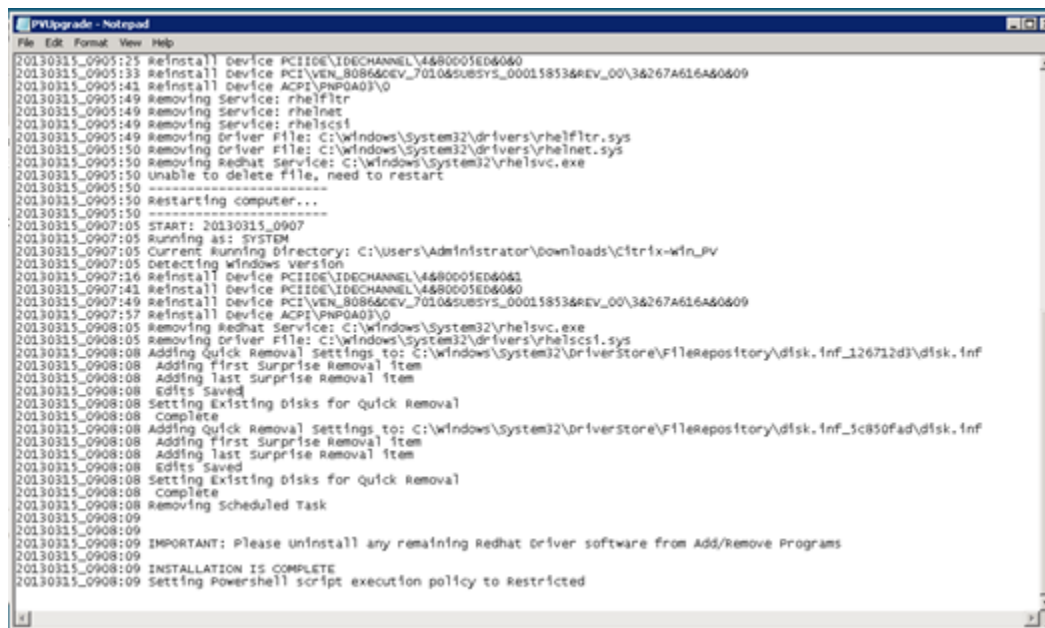
If you do not see the uninstaller dialog box, click **Red Hat Paravirtualiz...** in the Windows taskbar.



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.

```
Microsoft Windows NT 6.0.6002 Service Pack 2 (en-US)
Ec2Config service v2.1.9.0
RedHat PV NIC Driver v1.3.10.0
2013/03/15 17:11:01Z: Waiting for meta-data accessibility...
2013/03/15 17:11:02Z: Meta-data is now available.
<RDPCERTIFICATE>
<THUMBPRINT>D6BFD64F21359516C781CA7DF2821C5EFC35648A</THUMBPRINT>
</RDPCERTIFICATE>
<Username>Administrator</Username>
<Password>
!79ThJFFdLyIL38IZht0FBrjet3vnT2csTiU/XGVmRCH7kQcBznAnXrKd1srXlX19Bw/Med9b38jFJqv01IUpgNNJRZoCdc7Ib0w
</Password>
2013/03/15 17:11:30Z: Product activation was successful.
2013/03/15 17:11:32Z: Message: Windows is Ready to use
Microsoft Windows NT 6.0.6002 Service Pack 2 (en-US)
Ec2Config service v2.1.9.0
2013/03/15 21:04:24Z: There was an exception writing driver information to console: System.Exception:
at Ec2Config.Service1.Go()
2013/03/15 21:04:35Z: Waiting for meta-data accessibility...
2013/03/15 21:04:40Z: Meta-data is now available.
<RDPCERTIFICATE>
<THUMBPRINT>D6BFD64F21359516C781CA7DF2821C5EFC35648A</THUMBPRINT>
</RDPCERTIFICATE>
2013/03/15 21:05:08Z: Product activation was successful.
2013/03/15 21:05:09Z: Message: Windows is Ready to use
Microsoft Windows NT 6.0.6002 Service Pack 2 (en-US)
Ec2Config service v2.1.9.0
Citrix PV Ethernet Adapter v5.9.960.49119
2013/03/15 21:07:20Z: Waiting for meta-data accessibility...
2013/03/15 21:07:21Z: Meta-data is now available.
<RDPCERTIFICATE>
<THUMBPRINT>D6BFD64F21359516C781CA7DF2821C5EFC35648A</THUMBPRINT>
</RDPCERTIFICATE>
2013/03/15 21:07:27Z: Message: Windows is Ready to use
```

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`.



```
PVUpgrade - Notepad
File Edit Format View Help
20130315_0905:25 reinstall Device PCI\IDE\IDECHANNEL\4480005ED6060
20130315_0905:33 reinstall Device PCI\VEN_B0864CEV_7010&SUBSYS_0001583&REV_00\3&267A616A&0609
20130315_0905:41 reinstall Device ACPI\PNP0A03\0
20130315_0905:49 removing Service: rhelnet
20130315_0905:49 removing Service: rhelnet
20130315_0905:49 removing Service: rhelnet
20130315_0905:49 Removing driver File: C:\windows\System32\drivers\rhelnet.sys
20130315_0905:50 Removing driver File: C:\windows\System32\drivers\rhelnet.sys
20130315_0905:50 Removing Redhat Service: C:\windows\System32\rhelnet.exe
20130315_0905:50 unable to delete file, need to restart
20130315_0905:50
20130315_0905:50 Restarting computer...
20130315_0905:50
20130315_0907:05 START: 20130315_0907
20130315_0907:05 Running as: SYSTEM
20130315_0907:05 Current Running Directory: C:\Users\Administrator\downloads\Citrix-win_PV
20130315_0907:05 Detecting windows version
20130315_0907:16 reinstall Device PCI\IDE\IDECHANNEL\4480005ED6060
20130315_0907:41 reinstall Device PCI\IDE\IDECHANNEL\4480005ED6060
20130315_0907:49 reinstall Device PCI\VEN_B0864CEV_7010&SUBSYS_0001583&REV_00\3&267A616A&0609
20130315_0907:57 reinstall Device ACPI\PNP0A03\0
20130315_0908:05 Removing Redhat Service: C:\windows\System32\rhelnet.exe
20130315_0908:05 Removing driver File: C:\windows\System32\drivers\rhelnet.sys
20130315_0908:08 Adding Quick Removal Settings to: C:\windows\System32\DriverStore\FileRepository\disk_inf_126712d3\disk_inf
20130315_0908:08 Adding First Surprise Removal Item
20130315_0908:08 Adding Last Surprise Removal Item
20130315_0908:08 Edits Saved
20130315_0908:08 Setting Existing disks for quick Removal
20130315_0908:08 Adding Quick Removal Settings to: C:\windows\System32\DriverStore\FileRepository\disk_inf_3c850fad\disk_inf
20130315_0908:08 Adding First Surprise Removal Item
20130315_0908:08 Adding Last Surprise Removal Item
20130315_0908:08 Edits Saved
20130315_0908:08 Setting Existing disks for quick Removal
20130315_0908:08 complete
20130315_0908:08 Removing Scheduled Task
20130315_0908:09
20130315_0908:09
20130315_0908:09 IMPORTANT: Please uninstall any remaining Redhat Driver software from Add/Remove Programs
20130315_0908:09
20130315_0908:09 INSTALLATION IS COMPLETE
20130315_0908:09 Setting Powershell script execution policy to Restricted
```

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 \(p. 62\)](#). 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 \(p. 138\)](#).
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 \(p. 62\)](#). 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 \(p. 151\)](#).

To upgrade a Windows Server 2003 AMI

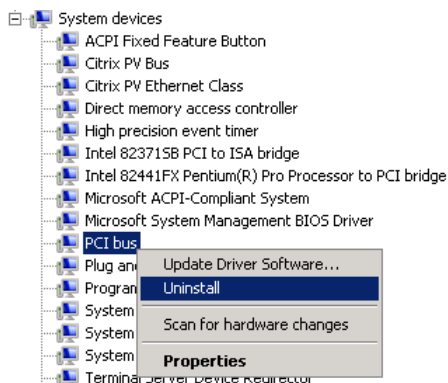
1. Connect to your instance and log in as the local administrator. For more information about connecting to your instance, see [Connecting to Your Windows Instance Using RDP \(p. 138\)](#).
2. In your instance, download the Citrix upgrade package by going to [Amazon EC2 Windows Paravirtual Driver Upgrade 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

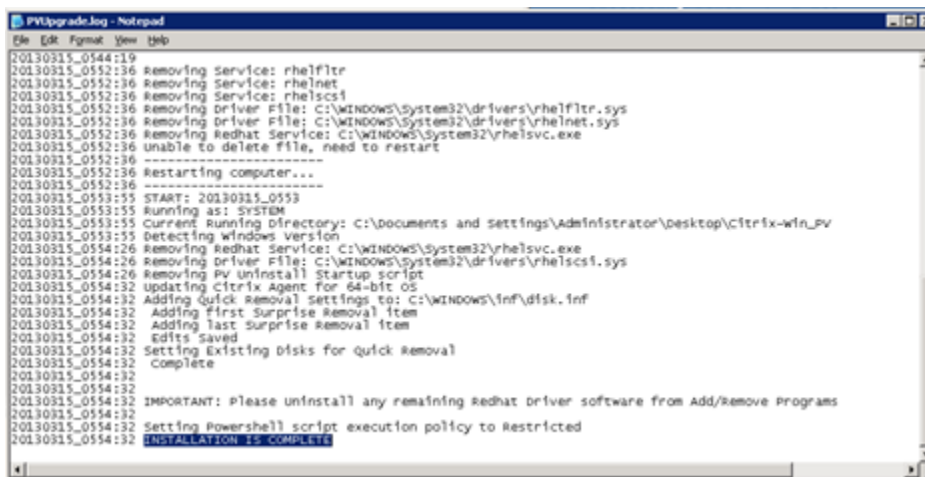
If you do not see the uninstaller dialog box, click **Red Hat Paravirtualiz...** in the Windows taskbar.



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

This topic addresses issues that you might encounter with the Citrix PV driver.

Contents

- [TCP Offloading \(p. 172\)](#)
- [Time Synchronization \(p. 174\)](#)

TCP Offloading

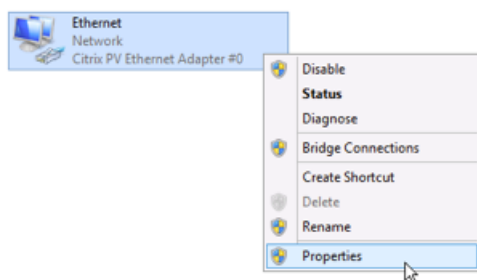
By default, TCP offloading is enabled for the Citrix PV drivers in Windows AMIs. If you encounter transport-level errors or packet transmission errors (as visible on the Windows Performance Monitor)—for example, when you're running certain SQL workloads—you may need to disable this feature.

Note

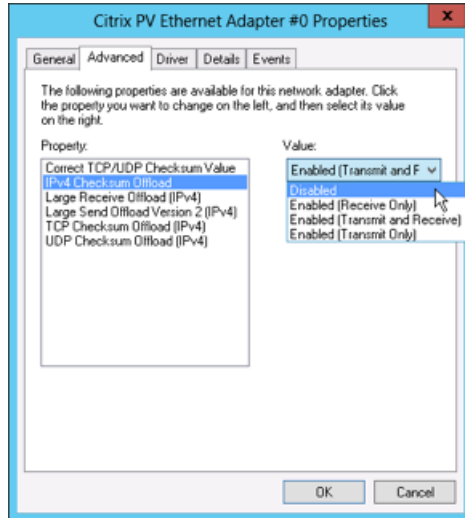
Disabling TCP offloading may reduce the network performance of your instance.

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)**



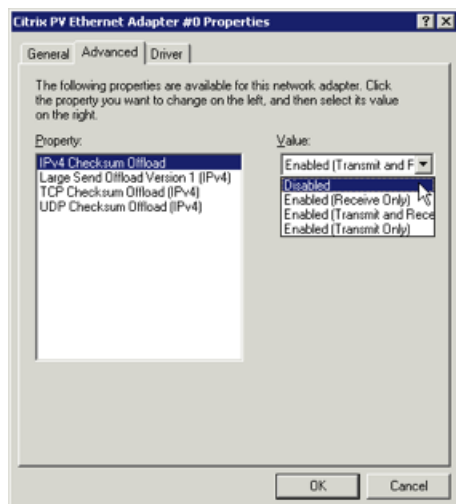
8. Click **OK**.
9. Run the following commands from a Command Prompt window.

```
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 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
```

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. 169).

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. 175)
- [Resetting an Administrator Password that's Lost or Expired](#) (p. 175)

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 \(p. 126\)](#).

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 \(p. 162\)](#).
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**.
 - c. In the **Stop Instances** dialog box, click **Yes, Stop**. After the instance has stopped, proceed with the next step.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Resetting an Administrator Password that's Lost or Expired**

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

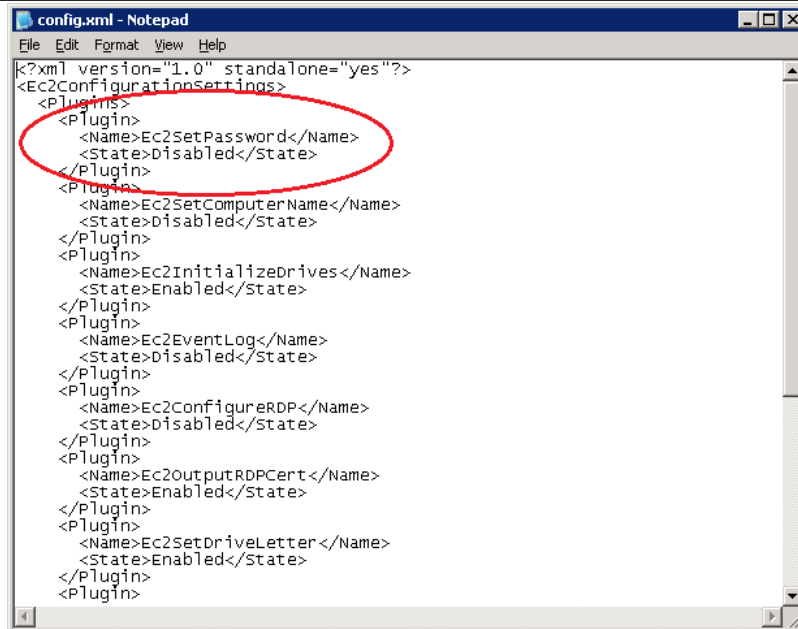
If your temporary instance is based on the same AMI that the original instance is based on, and the operating system is later than Windows Server 2003, you must complete additional steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision. 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 `xvdE` (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 [Make the Volume Available on Windows \(p. 363\)](#).

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.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Resetting an Administrator Password that's Lost or Ex-
pired**



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.)

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Resetting an Administrator Password that's Lost or Expired**

```
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.

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 \(p. 179\)](#)
- [Step 2: Configure a Secondary Private IP Address for Your Windows Instance \(p. 181\)](#)
- [Step 3: Configure Applications to Use the Secondary Private IP Address \(p. 182\)](#)

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 \(p. 325\)](#).
- 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 \(p. 328\)](#).

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.

Amazon Elastic Compute Cloud User Guide for Microsoft Windows

Step 1: Configure Static IP Addressing on Your Windows 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:

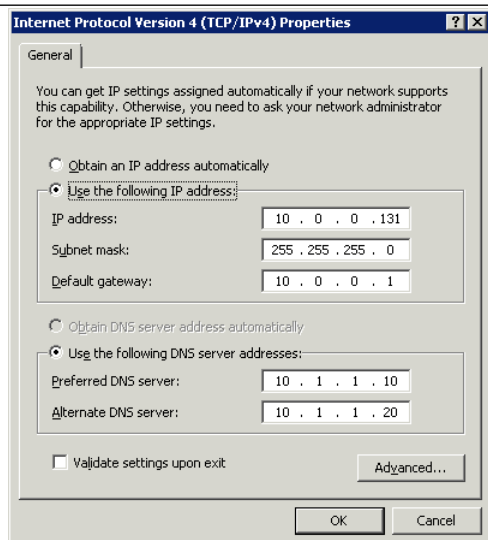
```
C:\> %SystemRoot%\system32\control.exe ncpa.cpl
```

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.

Step 2: Configure a Secondary Private IP Address for
Your Windows 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.

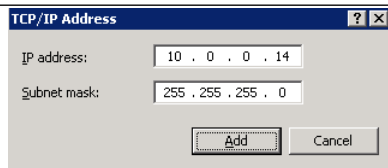
Step 2: Configure a Secondary Private IP Address for Your Windows Instance

After you have set up static IP addressing on your Windows instance, you are ready to prepare a second private IP address.

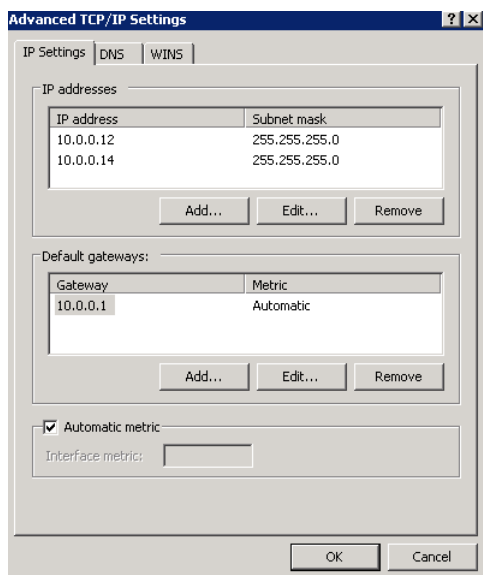
To configure a secondary IP address for a Windows instance

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. 179), and then click **Add**.

Step 3: Configure Applications to Use the Secondary
Private IP Address



13. Verify the IP address settings, and then click **OK**.



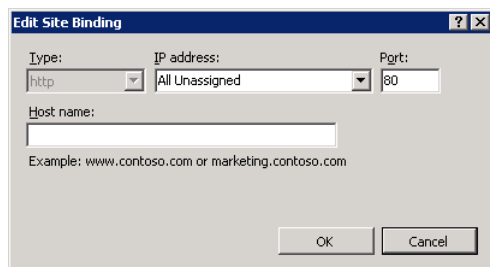
14. Click **OK** again, and then click **Close**.
15. To confirm that the secondary IP address has been added to the operating system, at a command prompt, run the command **ipconfig /all**.

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**.

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. 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 \(p. 183\)](#)
- [Configuring Network Time Protocol \(NTP\) \(p. 184\)](#)
- [Configuring Time Settings for Windows Server 2008 and later \(p. 184\)](#)
- [Configuring Time Settings for Windows Server 2003 \(p. 185\)](#)

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. 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
```

- (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.

- (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    0xffffffffc4
    DaylightName                        REG_SZ       @tzres.dll,-211
    DaylightStart                       REG_BINARY   00000300020002000000000000000000
    StandardBias                        REG_DWORD    0x0
    StandardName                        REG_SZ       @tzres.dll,-212
    StandardStart                       REG_BINARY   00000B00010002000000000000000000
    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 \(p. 164\)](#).

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.

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
CPU utilization	CPUUtilization (p. 201)	
Memory utilization		Monitoring Scripts for Amazon EC2 Instances (p. 249)
Memory used		Monitoring Scripts for Amazon EC2 Instances (p. 249)
Memory available		Monitoring Scripts for Amazon EC2 Instances (p. 249)

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Automated and Manual Monitoring**

Item to Monitor	Amazon EC2 Metric	Monitoring Script
Network utilization	NetworkIn (p. 201) NetworkOut (p. 201)	
Disk performance	DiskReadOps (p. 201) DiskWriteOps (p. 201)	
Disk Swap utilization Swap used (Linux)		Monitoring Scripts for Amazon EC2 Instances (p. 249)
Page File utilization Page File used Page File available		Monitoring Scripts for Amazon EC2 Instances (p. 249)
Disk Reads/Writes	DiskReadBytes (p. 201) DiskWriteBytes (p. 201)	
Disk Space utilization		Monitoring Scripts for Amazon EC2 Instances (p. 249)
Disk Space used		Monitoring Scripts for Amazon EC2 Instances (p. 249)
Disk Space available		Monitoring Scripts for Amazon EC2 Instances (p. 249)

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 \(p. 187\)](#)
- [Manual Monitoring Tools \(p. 188\)](#)

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 [Monitoring Instances with Status Checks \(p. 189\)](#).

- **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 [Monitoring Instances with Status Checks \(p. 189\)](#).

- **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 \(p. 197\)](#).
- **Amazon EC2 Monitoring Scripts** - Perl and PowerShell scripts that can monitor memory, disk, and page/swap file usage in your instances. For more information, see [Monitoring Scripts for Amazon EC2 Instances \(p. 249\)](#).
- **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 \(p. 449\)](#).

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

- [Monitoring Instances with Status Checks \(p. 189\)](#)
- [Monitoring Events for Your Instances \(p. 194\)](#)

Monitoring Instances with Status Checks

With instance status monitoring you can quickly determine whether Amazon EC2 has detected any problems that may prevent your instances from running applications. Amazon EC2 performs automated checks on every running Amazon 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 (pending, running, stopping, etc.) 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 \(p. 192\)](#).

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 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

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

Note

Status checks that occur during instance reboot or while a Windows instance store-backed instance is being bundled will report an instance status check failure until the instance becomes available again.

Viewing Status

AWS provides you with several ways to view and work with status checks: You can use the AWS Management Console, interact directly with the API, or use the command line interface.

Amazon EC2 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, click **Instances**.
3. On the **Instances** page, the **Status Checks** column lists the operational status of each instance.
4. To view an individual instance's status, select the instance, and then click the **Status Checks** tab.

Amazon Elastic Compute Cloud User Guide for Microsoft Windows

Monitoring Instances with Status Checks

The screenshot shows the AWS Management Console interface for monitoring instance status checks. At the top, there are tabs for 'Description', 'Status Checks' (which is selected), 'Monitoring', and 'Tags'. Below the tabs, a message states: 'Status checks detect problems that may impair this instance from running your applications. [Learn more](#) about status checks.' A button labeled 'Create Status Check Alarm' is visible. The main content is divided into two columns: 'System Status Checks' and 'Instance Status Checks'. Under 'System Status Checks', it says 'These checks monitor the AWS systems required to use this instance and ensure they are functioning properly.' and shows 'System reachability check passed' in green. Under 'Instance Status Checks', it says 'These checks monitor your software and network configuration for this instance.' and shows 'Instance reachability check failed at October 7, 2013 11:52:11 AM UTC+2 (16 minutes ago)' in red. At the bottom, there is a section for 'Additional Resources' with a link to 'Submit feedback' and a note about reporting issues.

Note

If you have an instance with a failed status check and the instance has been unreachable for over 20 minutes, you can click **Contact AWS Support** to submit a request for assistance.

Command Line Interface

To do this	Run this command
Get the status of all instances	<code>describe-instance-status</code>
Get the status of all instances with a instance status of impaired	<code>describe-instance-status --filters Name=instance-status.status,Values=impaired</code>
Get the status of a single instance with instance ID i-15a4417c	<code>describe-instance-status --instance-ids -i-15a4417c</code>

For more information about using the **describe-instance-status** command, see [describe-instance-status](#) in the *AWS Command Line Interface Reference*.

API

You can use the `DescribeInstanceStatus` action to retrieve the status of your instances. For more information, see [DescribeInstanceStatus](#) in the *Amazon Elastic Compute Cloud API Reference*.

Reporting Status

You can provide feedback about your instances if you are having problems with an instance whose status is not shown as impaired, or 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 reported via this form. Providing feedback does not change the status check results that you currently see for this instance.

If you are in need of technical assistance specific to your account, please post your question to the Developer Forums or contact Premium Support.

Amazon EC2 Console

To report status feedback using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, click **Instances**.
3. On the **Instances** page, click on the instance on which you want to report status.
4. Click the **Status Checks** tab, and then click **Submit feedback**.
5. Complete the information on the **Report Instance Status** page.

Command Line Interface

Use the `report-instance-status` command to send status feedback using the command line interface. The command uses the following syntax:

```
aws ec2 report-instance-status [--instances ...] [--status ...] [--reason-codes]
..]
```

For more information about using the `report-instance-status` command, see the [report-instance-status](#) command in the *AWS Command Line Interface Reference*.

API

You can use the `ReportInstanceStatus` action to submit feedback about a running instance's status. 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 Elastic Compute Cloud 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. To create or change these alarms, you can use either the AWS Management Console or the command line interface (CLI).

AWS Management Console

To create a status check alarm

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 when an instance fails an instance check or system status check.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Instances**.
3. Select an instance, and then on the **Status Checks** tab, click **Create Status Check Alarm**.
4. In the **Create Alarm** dialog box, select the **Send a notification to** check box, and then choose an existing Amazon Simple Notification Service (SNS) topic or create a new SNS topic to use for this alarm.
5. In the **With these recipients** box, type your email address (e.g., john.stiles@example.com) and the addresses of any additional recipients, separated by commas.
6. In the **Whenever** drop-down list, select the status check you want to be notified about (e.g., Status Check Failed (Any), Status Check Failed (Instance), or Status Check Failed (System)).
7. In the **For at least** box, set the number of periods you want to evaluate (for example, 2) and in the **consecutive periods** drop-down menu, select the evaluation period duration (for example, 5 minutes) before triggering the alarm and sending an email.
8. To change the default name for the alarm, in the **Name of alarm** box, type a friendly name for the alarm (for example, StatusCheckFailed), and then click **Create Alarm**.

Important

If you added an email address to the list of recipients or created a new topic, Amazon SNS will send a subscription confirmation email message to each new address shortly after you create an alarm. Remember to click the link contained in that message, which confirms your subscription. Alert notifications are sent only to confirmed addresses.

To edit a status check alarm

If you need to make changes to an instance status alarm, you can edit it.

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**, and then click **Add/Edit Alarms**.
4. In the **Alarm Details** dialog box, click the name of the alarm.
5. In the **Edit Alarm** dialog box, make the desired changes, and then click **Save**.

Command Line Interface

To create a status check alarm using the CLI

You can create a status check alarm using the AWS CLI. In the following example, the alarm publishes a notification to a specific SNS topic that has the ARN `arn:aws:sns:us-east-1:111111111111:StatusCheckNotifications` when instance `i-ab12345` fails either the instance check or system status check for at least two periods. (The metric is `StatusCheckFailed`.) For more information, see the [put-metric-alarm](#) command in the *AWS Command Line Interface Reference*.

1. At a command prompt, type `aws cloudwatch list-metrics` to view the list of all available Amazon CloudWatch metrics for the services in AWS that you're using.
2. In the list of metrics, review the Status Check metrics that have the **AWS/EC2** namespace. These are the status check metrics that you can use to create a status check alarm.
3. At the command prompt, enter the following command:

```
C:\> aws cloudwatch put-metric-alarm --alarm-name StatusCheckFailed-Alarm-
for-i-ab12345 --alarm-description "Alarm when StatusCheckFailed metric has
a value of one for two periods" --metric-name StatusCheckFailed --namespace
AWS/EC2 --statistic Maximum --dimensions Name=InstanceId,Value=i-ab12345
--period 300 --unit Count --evaluation-periods 2 --threshold 1 --comparison-
operator GreaterThanThreshold --alarm-actions arn:aws:sns:us-east-
1:111111111111:StatusCheckNotifications
```

Where:

The **--alarm-name** is the name of the alarm. This is required.

The **--alarm-description** is a friendly description of the alarm.

The **--metric-name** is one of the available status metrics (e.g., `StatusCheckFailed`, `StatusCheckFailed_Instance`, or `StatusCheckFailed_System`). This is required.

The **--namespace** is the metric's namespace (e.g., `AWS/EC2`). This is required.

The **--statistic** is one of the following values: `Average`, `Sum`, `Minimum`, or `Maximum`. This is required.

The **--dimensions** are associated with the metric (e.g., `InstanceId=i-ab12345`).

The `--period` is the time frame (in seconds) in which Amazon CloudWatch metrics are collected. In this example, you would enter 300, which is 60 seconds multiplied by 5 minutes. This is required.

The `--unit` is unit for the alarm's associated metric.

The `--evaluation-periods` is the number of consecutive periods for which the value of the metric must be compared to the threshold. This is required.

The `--threshold` is the value to which the metric will be compared (e.g., 1). This is required.

The `--alarm-actions` is the list of actions to perform when this alarm is triggered. Each action is specified as an Amazon Resource Number (ARN). In this example, we want the alarm to send us an email using Amazon SNS.

Note

You can find the ARN for the Amazon SNS topic that the alarm will use in the Amazon SNS console:

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/>.
2. In the navigation pane, under **My Topics**, select the topic you want the alarm to send mail to.
3. The ARN is located in the **Topic ARN** field on the **Topic Details** pane.

The `--unit` is the unit of the metric on which to alarm (e.g., Count).

Monitoring Events for Your Instances

Instance status describes specific events that AWS may schedule for your instances, such as a reboot or retirement. These scheduled events are not frequent. If one of your instances will be affected by a scheduled event, you'll receive an email prior to the scheduled event with details about the event, as well as a start and end date. You can also view scheduled events for your instance by using the Amazon EC2 console, API, or CLI. For more information, see [Viewing Scheduled Events \(p. 194\)](#).

There are different types of scheduled events:

- **Reboot:** A reboot can be either an instance reboot or a system reboot.
- **System maintenance:** An instance may be temporarily affected by network maintenance or power maintenance.
- **Instance retirement:** An instance that's scheduled for retirement will be stopped or terminated.
- **Instance stop:** An instance may need to be stopped in order to migrate it to new hardware.

If one of your instances is scheduled for any of the above events, you may be able to take actions to control the timing of the event, or to minimize downtime. For more information, see [Working with an Instance That Has a Scheduled Event \(p. 195\)](#).

Contents

- [Viewing Scheduled Events \(p. 194\)](#)
- [Working with an Instance That Has a Scheduled Event \(p. 195\)](#)

Viewing Scheduled Events

You can view scheduled events for your instances using the Amazon EC2 console, the command line interface (CLI), or the API.

Amazon EC2 Console

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**. You can see a list of all resources with events associated with them. You can filter by instance or volume, or by specific status types.
3. Alternatively, you can do the following to view upcoming scheduled events:
 - a. In the navigation pane, click the **EC2 Dashboard**.
 - b. Under **Scheduled Events**, you can see the events associated with your Amazon EC2 instances and volumes.

Command Line Interface and API

To view the status of your instances, use the `ec2-describe-instance-status` command, or the `DescribeInstanceStatus` API action.

Working with an Instance That Has a Scheduled Event

This section discusses tasks you can perform if your instance has a scheduled event. For all scheduled events, your course of action will differ depending on whether your instance's root device volume is an Amazon EBS volume or an instance store volume. You can determine the root device type for an instance by checking the value of the **Root device type** field in the details pane on the **Instances** page.

Instances Scheduled for Reboot

AWS may schedule instances for a reboot in order to perform tasks such as applying patches, upgrades, or maintenance to the underlying host. There are two types of reboot events: system reboot and instance reboot. During a system reboot, your instance and the hardware supporting your instance is rebooted. During an instance reboot, your instance is rebooted, but the hardware supporting your instance is not rebooted. You can find out which type of reboot event is scheduled for your instance by using the Amazon EC2 console.

To view the type of scheduled reboot events

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Events**.
3. Select **Instance resources** from the filter list, and locate your instance.
4. Look under the **Event Type** column. The column should indicate `system-reboot` or `instance-reboot`.

Scheduled reboot events start within their scheduled maintenance window and typically complete in a matter of minutes. After a reboot completes, your instance is available to use; it is not necessary to wait until the scheduled end time.

To verify that the reboot has occurred, check your scheduled events and verify that the instance no longer shows a scheduled event. We recommend that you check your instance after it is rebooted to ensure that your application is functioning as you expected.

Actions Required for Instances Scheduled for Reboot

No action is required on your part if one of your instances is scheduled for reboot. We recommend that you wait for the reboot to occur automatically within its scheduled maintenance window.

If you choose, you may perform the reboot yourself to control the timing of the event.

Instance Reboot

For instance reboot events, you can perform the reboot yourself (be careful to not shut down or terminate your instance) at any time before the scheduled reboot event begins. For more information, see [Reboot Your Instance \(p. 143\)](#). After you reboot, any pending maintenance to the underlying host is performed automatically, and you can begin using your instance again after the instance has fully booted.

Note

After you perform the reboot, the scheduled event for the instance reboot is canceled immediately. The event's description is updated in the Amazon EC2 console to reflect this.

System Reboot

If your instance's root device is an Amazon EBS volume, you can stop and start it. For more information, see [Stop and Start Your Instance \(p. 140\)](#). If your instance's root device is an instance store volume, you can terminate it and launch a replacement.

Important

Any data stored on an instance store volume is lost when the instance is stopped or terminated, and cannot be recovered. This includes instance store volumes that are attached to an instance that has an Amazon EBS volume as the root device.

Instances Scheduled to Be Stopped or Retired

An instance is scheduled to be stopped or retired when AWS detects irreparable failure of the underlying hardware hosting your instance. When an instance reaches its scheduled retirement date, it is stopped or terminated by AWS. If your instance's root device is an Amazon EBS volume, the instance is stopped, and you can start it again at any time. If your instance's root device is an instance store volume, the instance is terminated, and cannot be used again.

Actions Required for Instances Scheduled to Be Stopped or Retired

If your instance's root device is an Amazon EBS volume, you can wait for the instance to be stopped or retired at the scheduled event start time. Alternatively, you can stop and start the instance yourself. Doing so migrates your instance to new hardware and help reduce unforeseen downtime. For more information about stopping your instance, as well as information about changes to your instance configuration when it's stopped, see [Stop and Start Your Instance \(p. 140\)](#).

If your instance's root device is an instance store volume, we recommend that you launch a replacement instance from your most recent AMI, and migrate all necessary data to the replacement instance before the scheduled retirement. You can then terminate the instance, or wait for it to be automatically terminated when it's retired.

For more information about instances scheduled for retirement and how to manage them, see [Instance Retirement \(p. 143\)](#).

Important

Any data stored on instance store volumes is lost when the instance is stopped or terminated, and cannot be recovered. This includes instance store volumes that are attached to an instance that has an Amazon EBS volume as the root device. Before the instance is stopped or terminated, ensure you retrieve any data from the instance store volume that you will need later.

Instances Scheduled for Maintenance

Instances are scheduled for maintenance when underlying Amazon EC2 hardware requires 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 offline for a brief period, and then rebooted.

Actions Required for Instances Scheduled for Maintenance

No action is required on your part if one of your instances is scheduled for maintenance. However, if you want to maintain normal operation during this time, you can launch a replacement instance from your most recent AMI, and migrate all necessary data to the replacement instance before the scheduled maintenance. Replacement instances are not affected by the same scheduled network or power maintenance.

For power maintenance, when a reboot is performed, all of your instance's configuration settings are retained.

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.
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 \(p. 198\)](#)
- [View Amazon EC2 Metrics \(p. 201\)](#)
- [Get Statistics for Metrics \(p. 207\)](#)
- [Graphing Metrics \(p. 223\)](#)
- [Create a CloudWatch Alarm \(p. 227\)](#)
- [Create Alarms That Stop or Terminate an Instance \(p. 234\)](#)

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**, 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**, 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 \(p. 129\)](#).

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-eafelb82"
    }
  ],
  "Instances": [
    {
      "Monitoring": {
        "State": "pending"
      },
      "PublicDnsName": null,
      "Platform": "windows",
      "State": {
        "Code": 0,
        "Name": "pending"
      },
      "EbsOptimized": false,
      "LaunchTime": "2014-02-24T18:02:49.000Z",
      "ProductCodes": [],

```


**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Enabling or Disabling Detailed Monitoring on an Amazon
EC2 Instance**

```
"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	<p>(Only valid for T2 instances) The number of CPU credits consumed during the specified period. This includes both credits earned during the period and credits withdrawn from the CPU credit balance for bursting.</p> <p>This metric identifies the amount of time during which physical CPUs were used for processing instructions by virtual CPUs allocated to the instance.</p> <p>Note CPU Credit metrics are available at a 5 minute frequency.</p> <p>Units: Count</p>
CPUCreditBalance	<p>(Only valid for T2 instances) The number of CPU credits that an instance has accumulated.</p> <p>This metric is used to determine how long an instance can burst beyond its baseline performance level at a given rate.</p> <p>Note CPU Credit metrics are available at a 5 minute frequency.</p> <p>Units: Count</p>
CPUUtilization	<p>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.</p> <p>Units: Percent</p>
DiskReadOps	<p>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. 366).</p> <p>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.</p> <p>Units: Count</p>

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
View Amazon EC2 Metrics**

Metric	Description
DiskWriteOps	<p>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. 366).</p> <p>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.</p> <p>Units: Count</p>
DiskReadBytes	<p>Bytes read from all ephemeral disks available to the instance (if your instance uses Amazon EBS, see Amazon EBS Metrics (p. 366).)</p> <p>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.</p> <p>Units: Bytes</p>
DiskWriteBytes	<p>Bytes written to all ephemeral disks available to the instance (if your instance uses Amazon EBS, see Amazon EBS Metrics (p. 366).)</p> <p>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.</p> <p>Units: Bytes</p>
NetworkIn	<p>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.</p> <p>Units: Bytes</p>
NetworkOut	<p>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.</p> <p>Units: Bytes</p>
StatusCheckFailed	<p>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.</p> <p>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.</p> <p>Units: Count</p>

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
View Amazon EC2 Metrics**

Metric	Description
StatusCheckFailed_Instance	<p>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.</p> <p>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.</p> <p>Units: Count</p>
StatusCheckFailed_System	<p>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.</p> <p>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.</p> <p>Units: Count</p>

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 <i>AutoScalingGroup</i> 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. Available for instances with Detailed Monitoring enabled.
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.

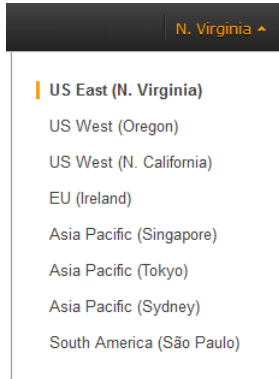
For more information about using the `GetMetricStatistics` action, see [GetMetricStatistics](#) in the *Amazon CloudWatch API Reference*.

AWS Management Console

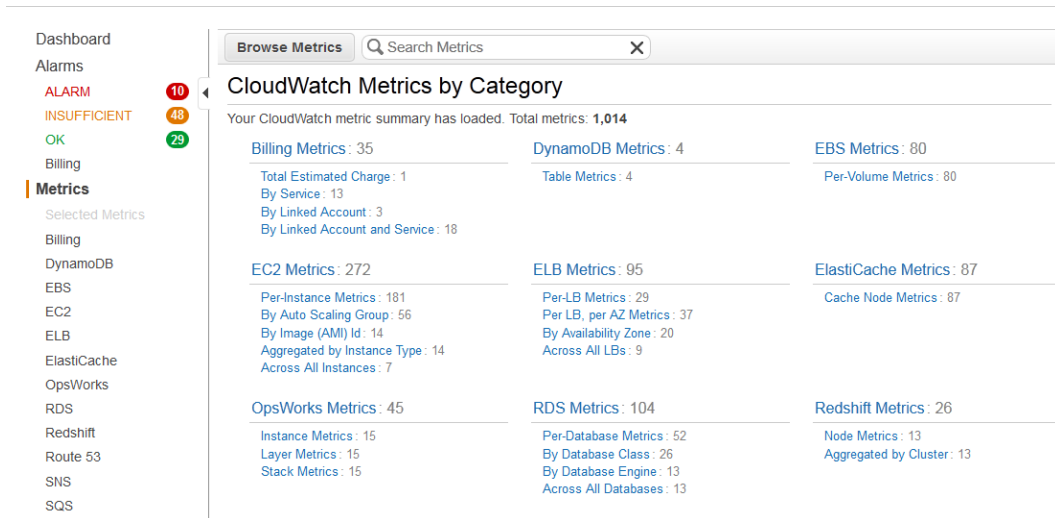
To view available metrics by category

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 Amazon 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.

Amazon Elastic Compute Cloud User Guide for Microsoft Windows

View Amazon EC2 Metrics

Browse Metrics Search Metrics EC2 > Per-Instance Metrics 1 to 50 of 181 Metrics

Showing all results (181) for EC2 > Per-Instance Metrics.

Select All | Clear

EC2 > Per-Instance Metrics

Instanceld	Metric Name
<input type="checkbox"/> i-14d5ac6c	CPUUtilization
<input type="checkbox"/> i-14d5ac6c	DiskReadBytes
<input type="checkbox"/> i-14d5ac6c	DiskReadOps
<input type="checkbox"/> i-14d5ac6c	DiskWriteBytes
<input type="checkbox"/> i-14d5ac6c	DiskWriteOps

Update Graph

Time Range

Relative Absolute UTC (GMT)

From: 12 hours ago

To: 0 minutes ago

Zoom: 1h | 3h | 6h | 12h | 1d | 3d | 1w | 2w

Select a metric above to view graph

Click a checkbox to select a metric
Click on text to add to search

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"
},
```

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
View Amazon EC2 Metrics**

```
{
  "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 \(p. 207\)](#)
- [Aggregating Statistics Across Instances \(p. 211\)](#)
- [Get Statistics Aggregated by Auto Scaling Group \(p. 216\)](#)
- [Get Statistics Aggregated by Image \(AMI\) ID \(p. 219\)](#)

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.

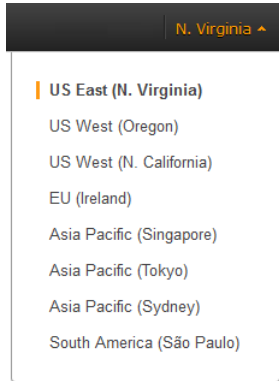
AWS Management Console

To display the average CPU utilization for a specific instance

1. Open the Amazon 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](#).

Amazon Elastic Compute Cloud User Guide for Microsoft Windows

Get Statistics for Metrics

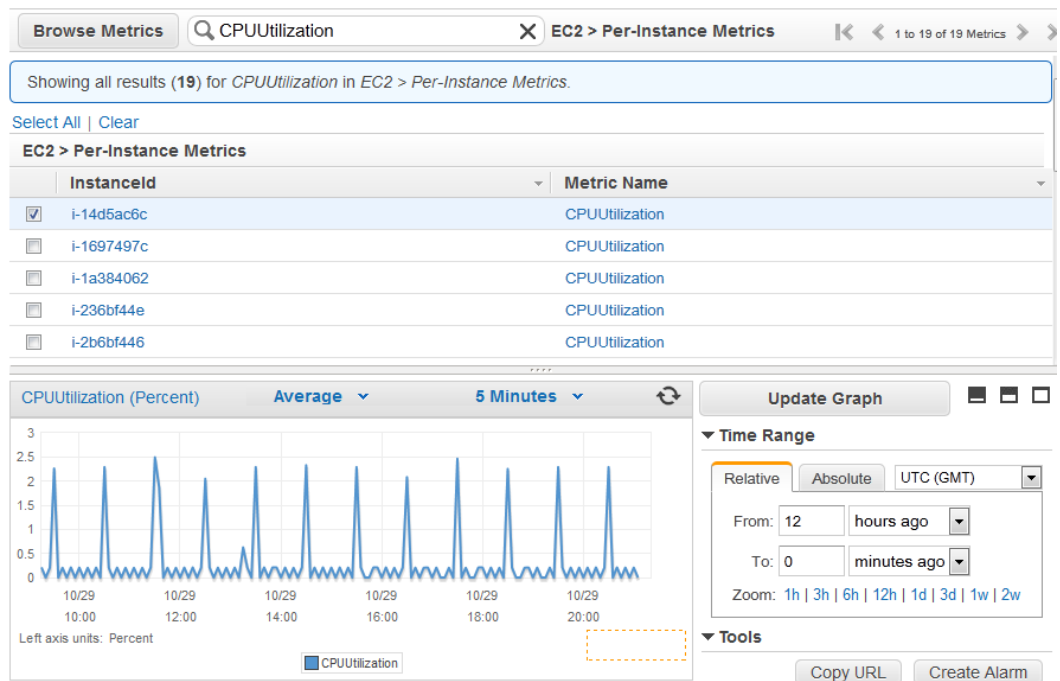


3. In the navigation pane, click **Metrics**.
4. In the **CloudWatch Metrics by Category** pane, select **EC2: Metrics**.

The metrics available for individual instances appear in the upper pane.

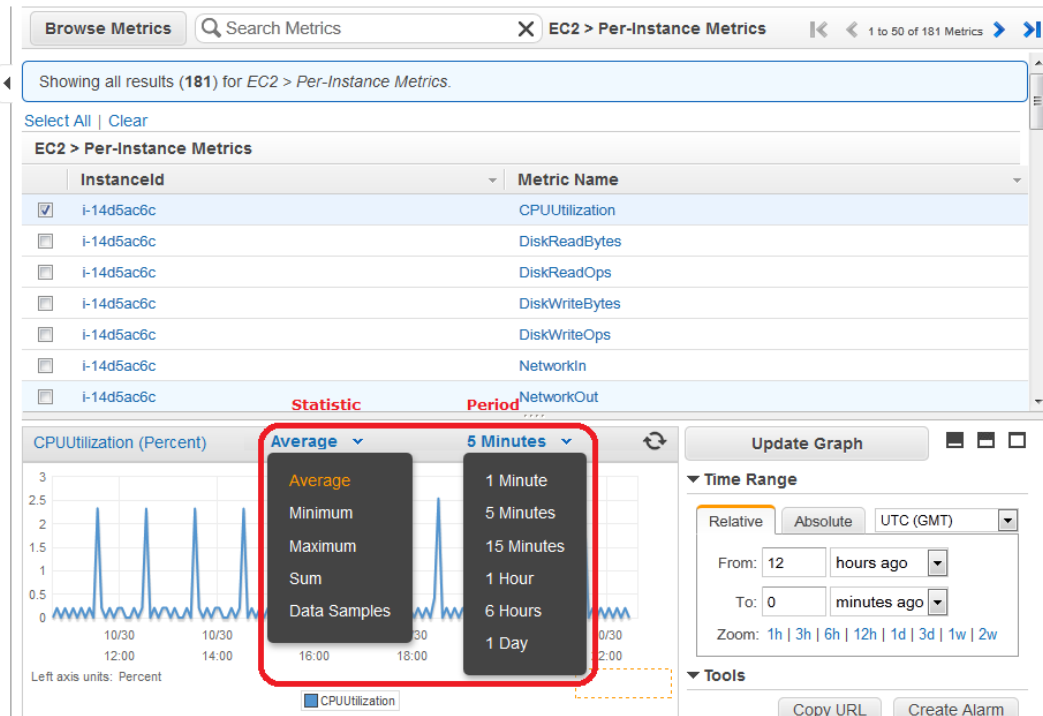
5. Select a row that contains **CPUUtilization** for a specific InstanceId.

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.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Get Statistics for Metrics**



- 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": 0.33300000000000002,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-19T03:18:00Z",
      "Maximum": 99.670000000000002,
      "Unit": "Percent"
    }
  ]
}
```

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Get Statistics for Metrics**

```
    },
    {
      "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](#).

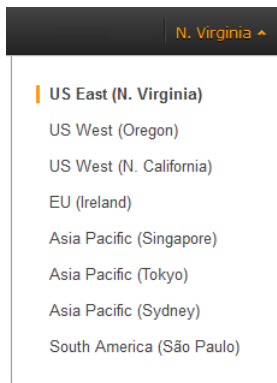
Important

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 Amazon 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**.

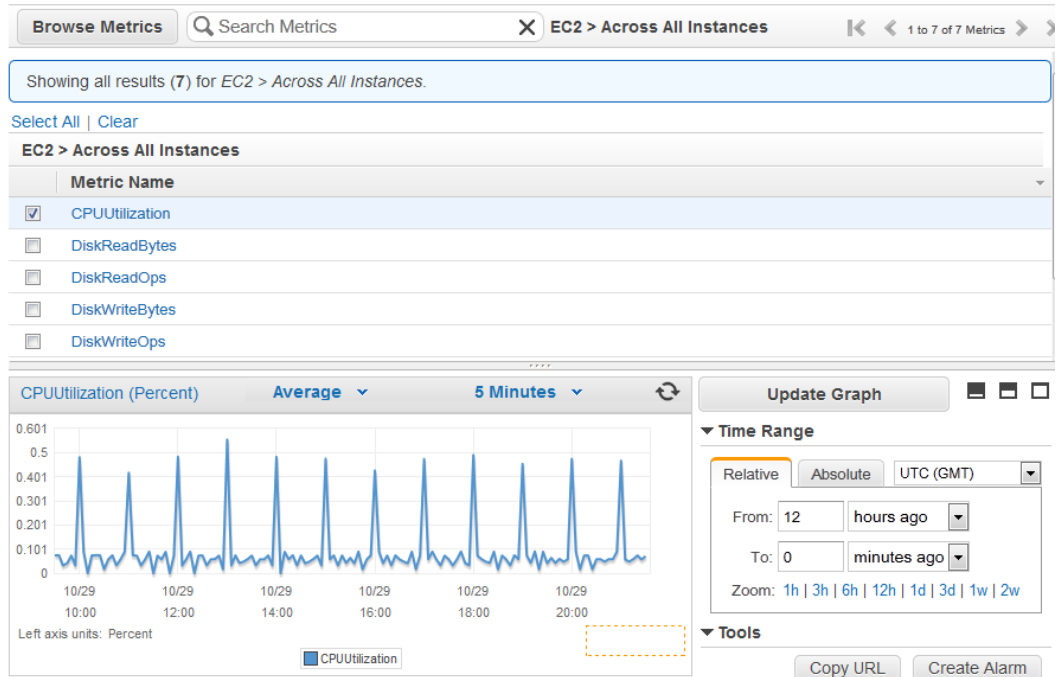
The metrics available across all instances are displayed in the upper pane.

5. In the upper pane, select the row that contains **CPUUtilization**.

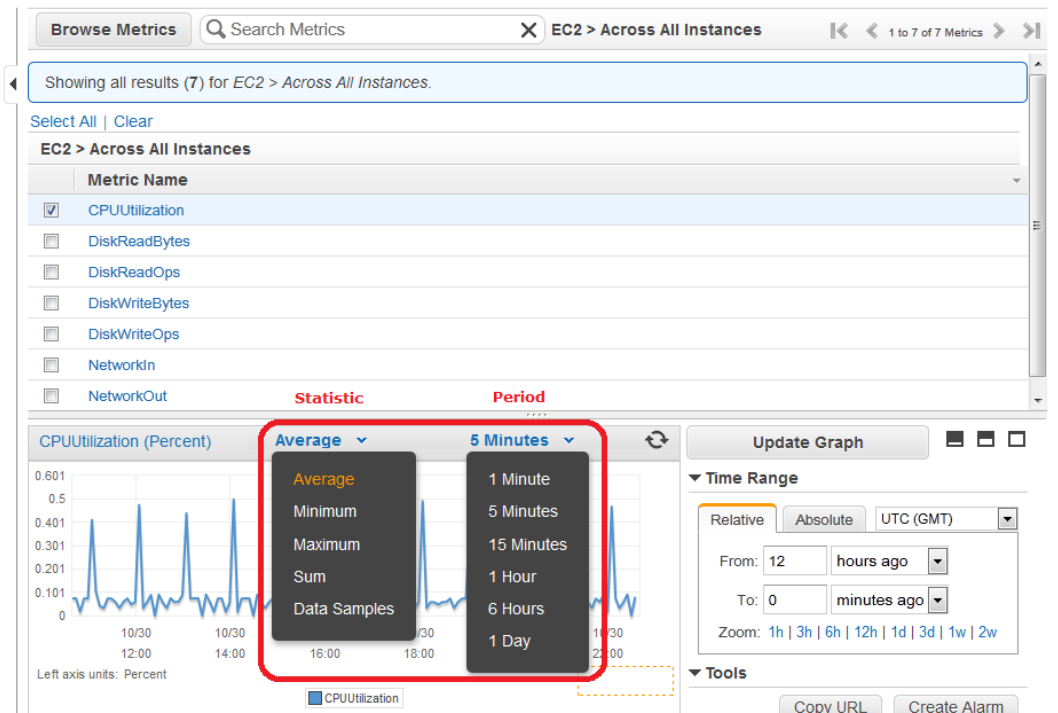
A graph showing `CPUUtilization` for your EC2 instances is displayed in the details pane.

Amazon Elastic Compute Cloud User Guide for Microsoft Windows

Get Statistics for Metrics



- To change the **Statistic**, e.g., Average, for the metric, choose a different value from the pop-up list.



- 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"
    }
  ]
}
```

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Get Statistics for Metrics**

```
    },
    {
      "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": 0.042875000000000001,  
        "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

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

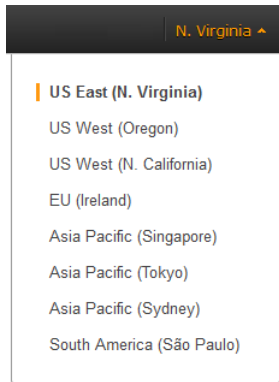
Start and end times must be within the last 14 days.

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 Amazon 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](#).

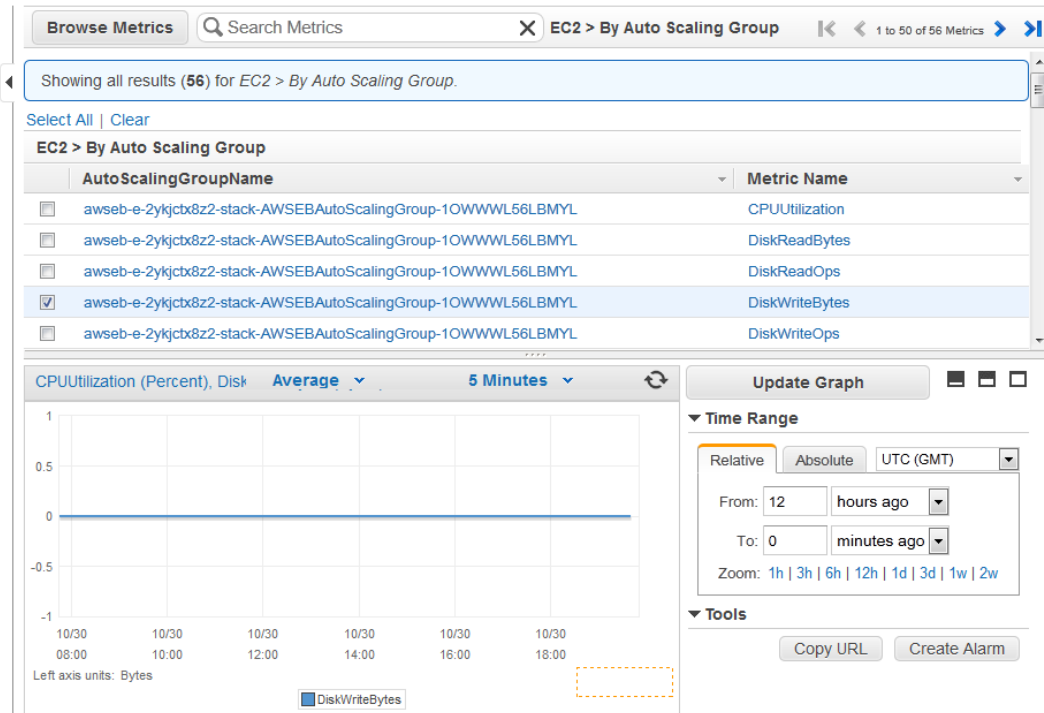


3. In the navigation pane, click **Metrics**.
4. In the **CloudWatch Metrics by Category** pane, under **EC2 Metrics**, select **By Auto Scaling Group**.

The metrics available for Auto Scaling groups are displayed in the upper pane.

5. Select the row that contains **DiskWriteBytes**.

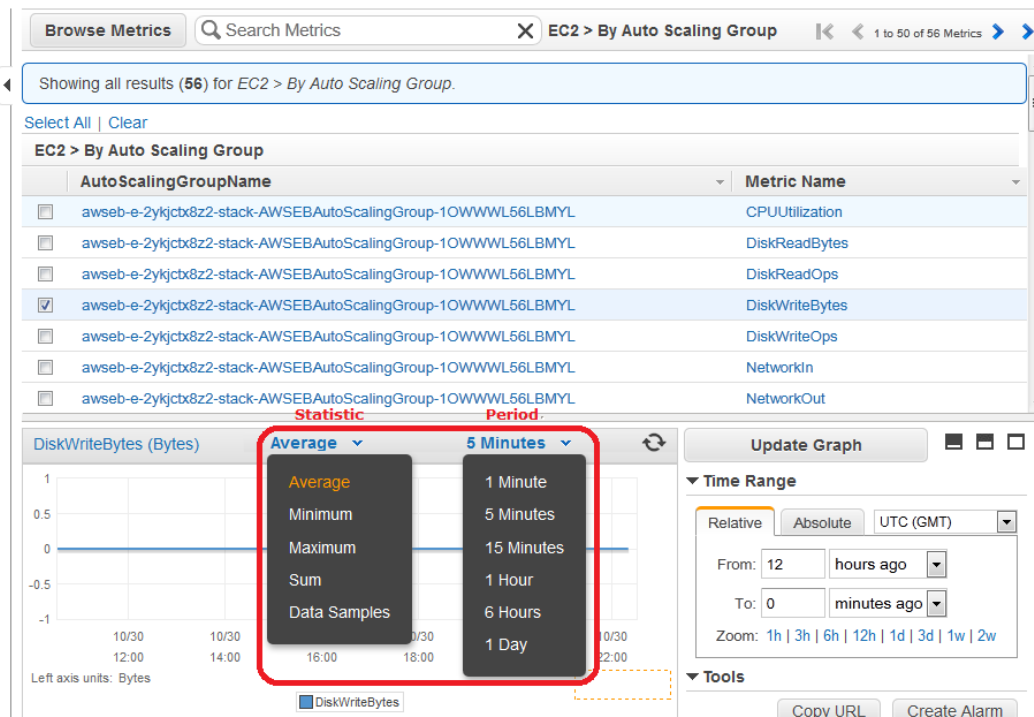
A graph showing `DiskWriteBytes` for all EC2 instances appears in the details pane.



Amazon Elastic Compute Cloud User Guide for Microsoft Windows

Get Statistics for Metrics

- To change the **Statistic**, e.g., Average, for the metric, choose a different value from the pop-up list.



- 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

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

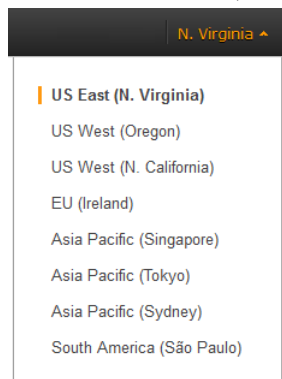
Start and end times must be within the last 14 days.

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 Amazon 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**.

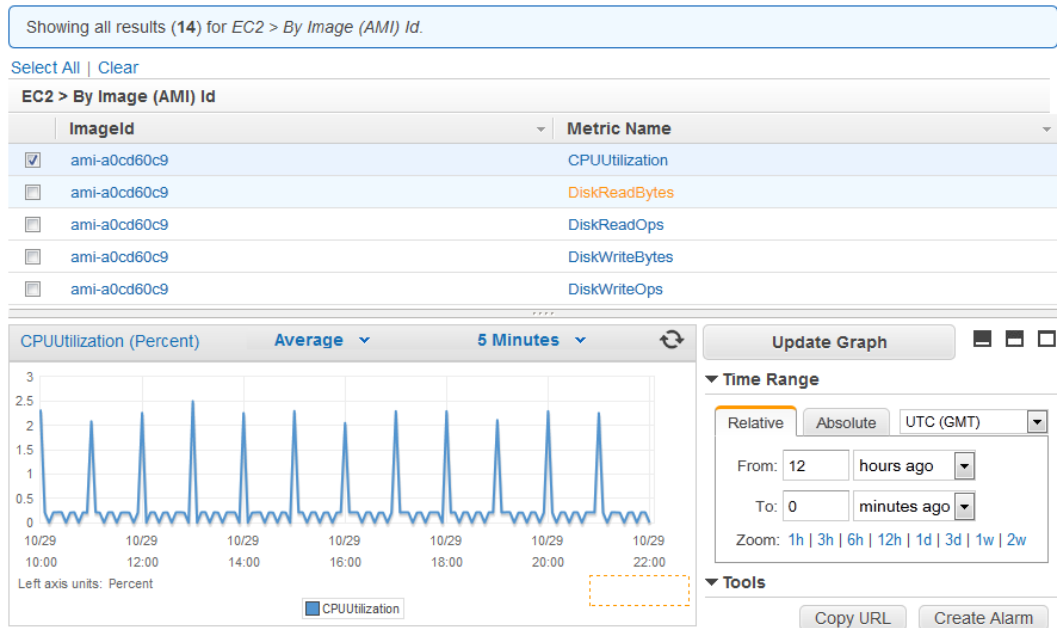
The metrics available for image IDs appear in the upper pane.

5. Select a row that contains **CPUUtilization** and an image ID.

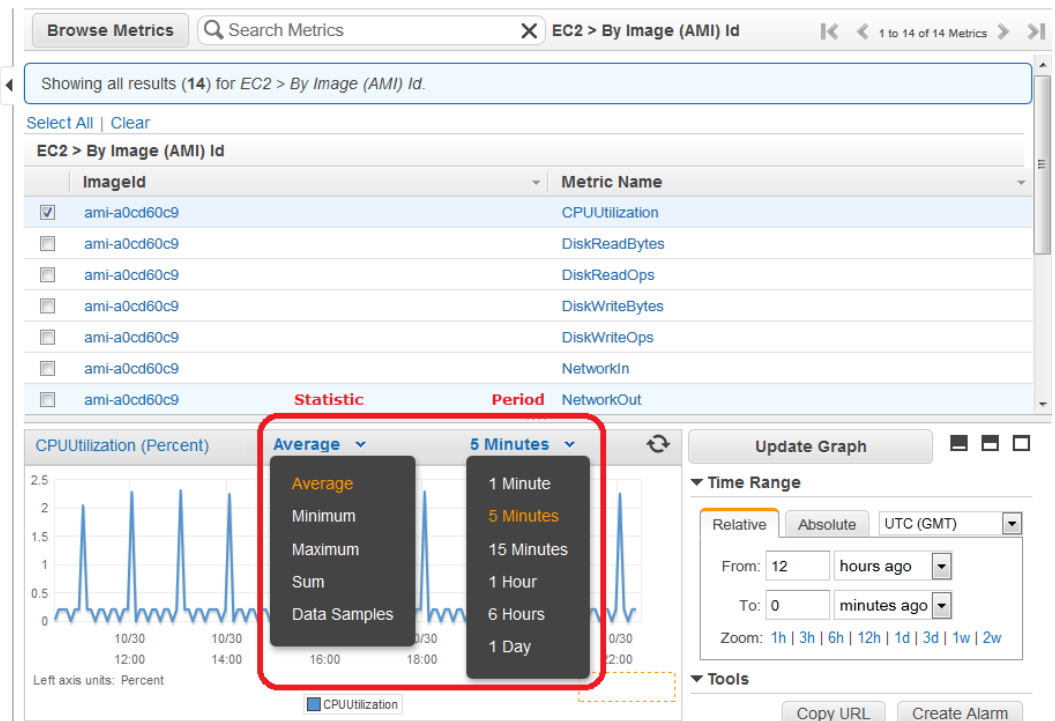
A graph showing average CPU utilization for all EC2 instances based on the `ami-c5e40dac` image ID appears in the details pane.

Amazon Elastic Compute Cloud User Guide for Microsoft Windows

Get Statistics for Metrics



- To change the **Statistic**, e.g., Average, for the metric, choose a different value from the pop-up list.



- 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

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Get Statistics for Metrics**

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": 0.0360000000000000011,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T13:00:00Z",
      "Average": 0.0376250000000000013,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T18:00:00Z",
      "Average": 0.0427500000000000003,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T21:00:00Z",
      "Average": 0.039705882352941188,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T20:00:00Z",
      "Average": 0.0393750000000000007,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T02:00:00Z",
      "Average": 0.041041666666666671,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T01:00:00Z",
      "Average": 0.041083333333333354,
      "Unit": "Percent"
    }
  ]
}
```

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Get Statistics for Metrics**

```
{
  "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)

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

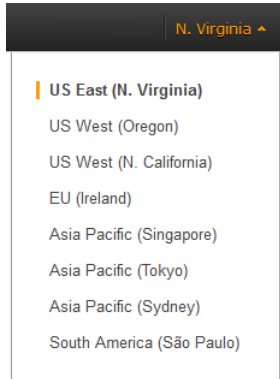
- [Graph a Metric \(p. 224\)](#)
- [Graph a Metric Across Resources \(p. 225\)](#)

Graph a Metric

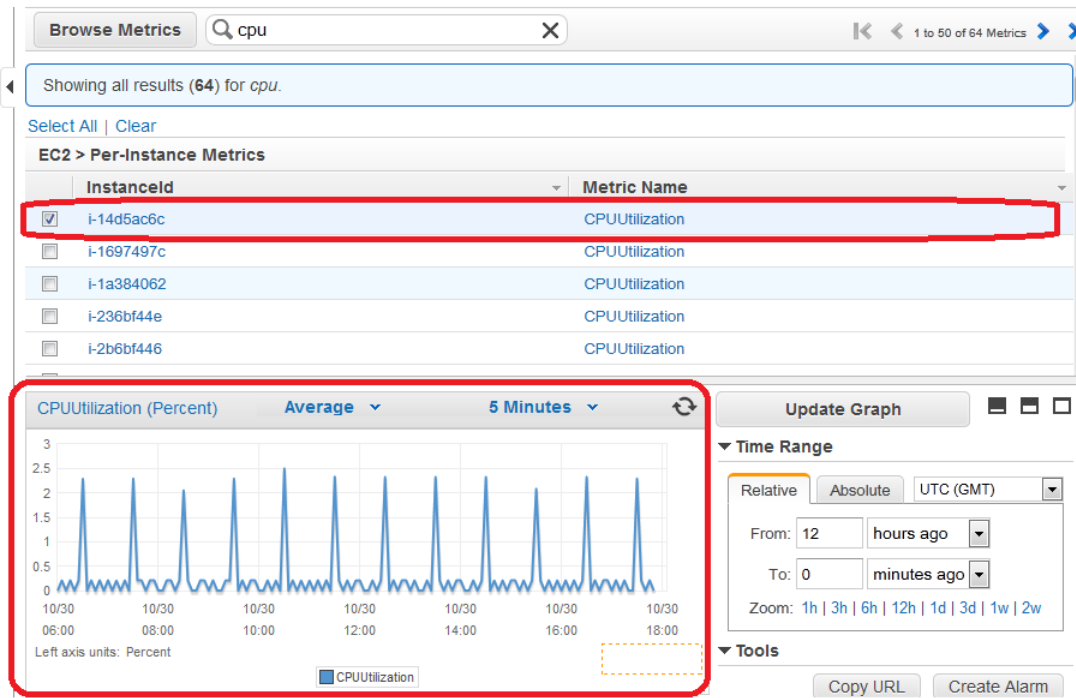
You can select a metric and create a graph of the data in CloudWatch. For example, you can select the CPUUtilization metric for an Amazon EC2 instance and display a graph of CPU usage over time for that instance.

To graph a metric

1. Open the Amazon 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.



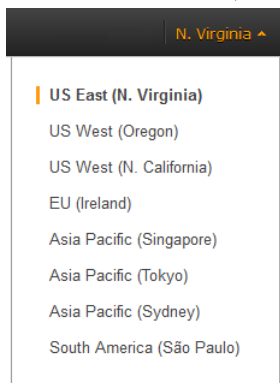
- To save this graph and access it later, in the details pane, under **Tools**, click **Copy URL**, and then in the **Copy Graph URL** dialog box, select the URL and paste it into your browser.

Graph a Metric Across Resources

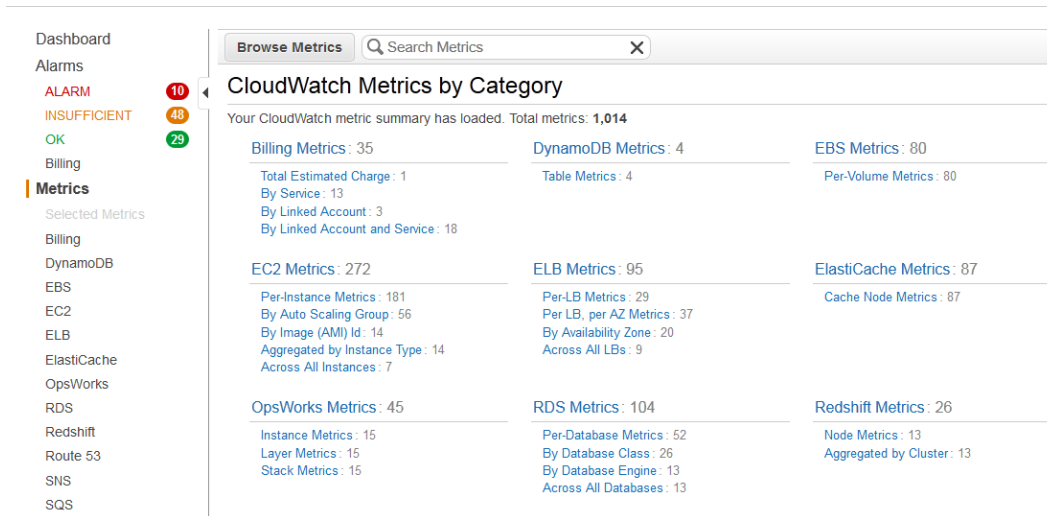
You can graph a metric across all resources to see everything on one graph. For example, you can graph the CPUUtilization metric for all Amazon EC2 instances on one graph.

To graph a metric across resources

- Open the Amazon CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
- If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see [Regions and Endpoints](#).

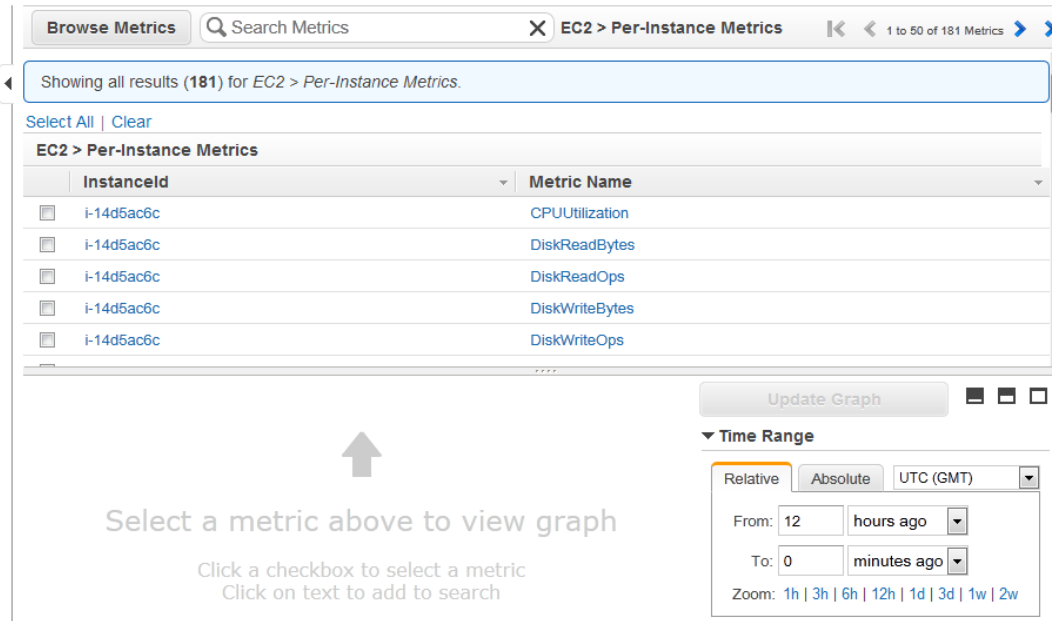


- In the navigation pane, click **Metrics**.



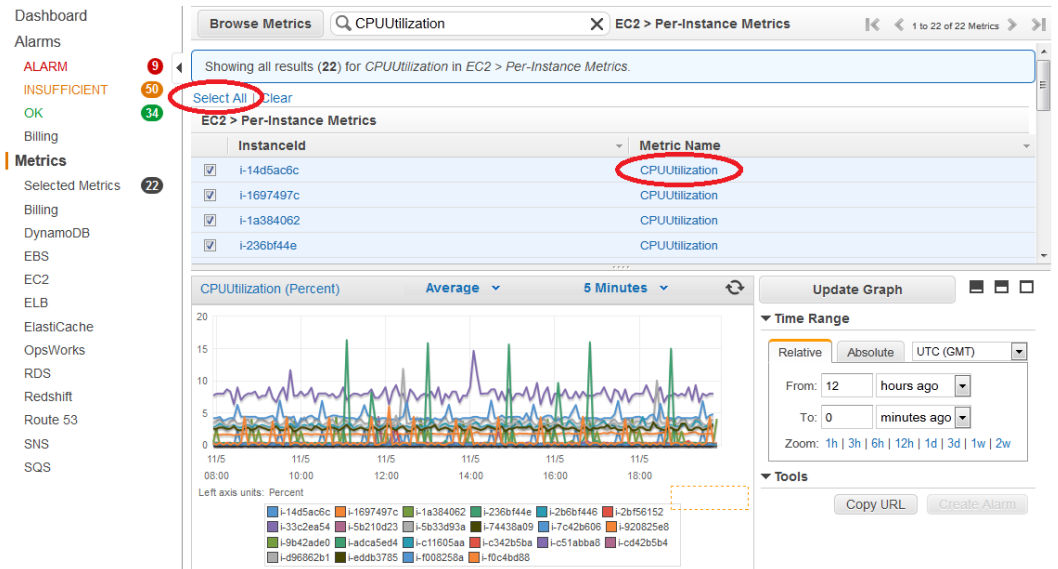
- In the **CloudWatch Metrics by Category** pane, select a metric category. For example, under **EC2 Metrics**, select **Per-Instance Metrics**.

Amazon Elastic Compute Cloud User Guide for Microsoft Windows Graphing 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 \(p. 227\)](#)
- [Send Email Based on Load Balancer Alarm \(p. 229\)](#)
- [Send Email Based on Storage Throughput Alarm \(p. 231\)](#)

Send Email Based on CPU Usage Alarm

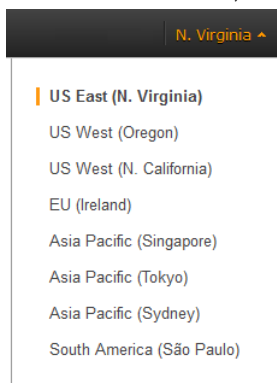
This scenario walks you through how to use the AWS Management Console or the command line interface to create an Amazon CloudWatch alarm that sends an Amazon Simple Notification Service email message when the alarm changes state from OK to ALARM.

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

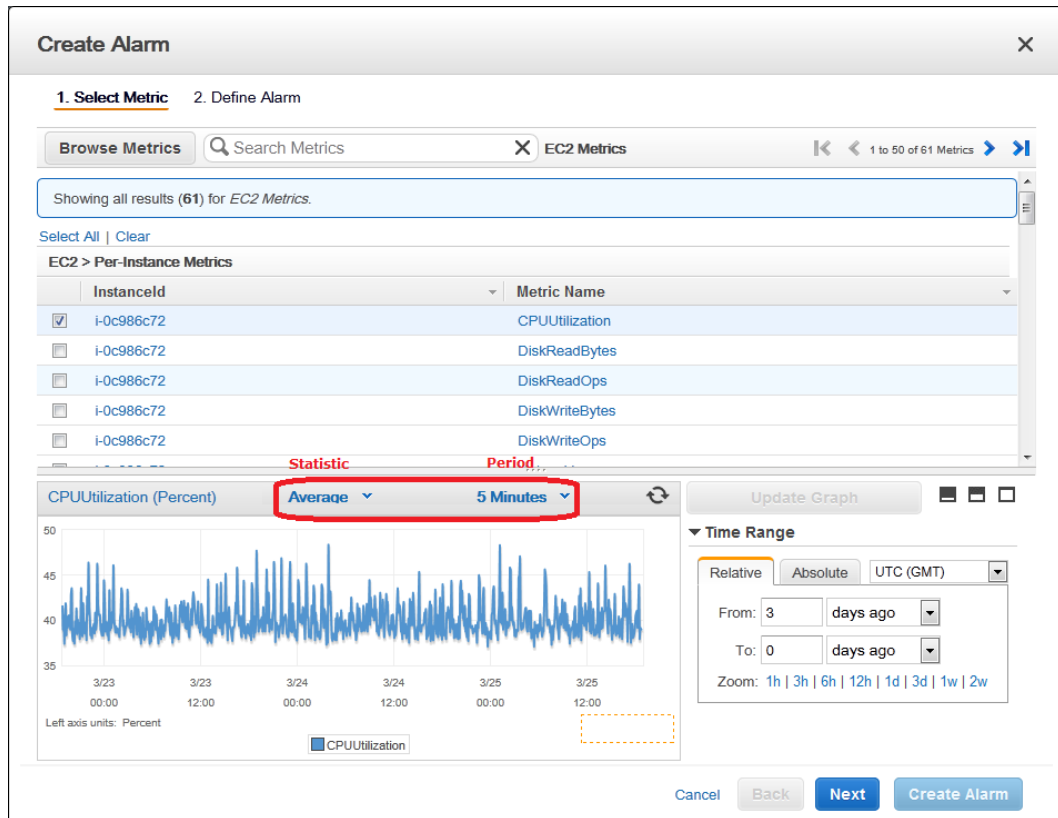
1. Open the Amazon CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see [Regions and Endpoints](#).



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.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Create a CloudWatch Alarm**



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: **myHigh-CpuAlarm**, 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

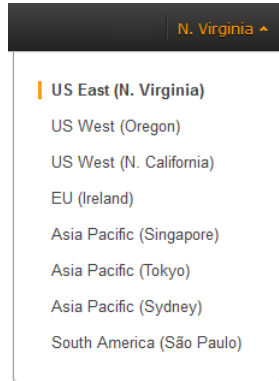
This scenario walks you through how to use the AWS Management Console or the command line interface to set up an Amazon Simple Notification Service notification and configure an alarm that monitors load balancer latency exceeding 100 ms.

AWS Management Console

To create a load balancer alarm that sends email

1. Open the Amazon 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](#).

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Create a CloudWatch Alarm**



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.

Create Alarm [X]

1. **Select Metric** 2. Define Alarm

Browse Metrics Search Metrics X ELB Metrics << 1 to 18 of 18 Metrics >>

<input checked="" type="checkbox"/>	awseb-e-2-AWSEBLoa-1K2GC5PYQHIV	Latency
<input type="checkbox"/>	awseb-e-2-AWSEBLoa-1K2GC5PYQHIV	RequestCount
<input type="checkbox"/>	awseb-e-2-AWSEBLoa-1K2GC5PYQHIV	SurgeQueueLength
<input type="checkbox"/>	awseb-e-2-AWSEBLoa-1K2GC5PYQHIV	UnHealthyHostCount
<input type="checkbox"/>	awseb-e-3-AWSEBLoa-1JBBEL2SOV385	HTTPCode_Backend_2XX
<input type="checkbox"/>	awseb-e-3-AWSEBLoa-1JBBEL2SOV385	HTTPCode_Backend_4XX
<input type="checkbox"/>	awseb-e-3-AWSEBLoa-1JBBEL2SOV385	HTTPCode_Backend_5XX
<input type="checkbox"/>	awseb-e-3-AWSEBLoa-1JBBEL2SOV385	HealthyHostCount
<input type="checkbox"/>	awseb-e-3-AWSEBLoa-1JBBEL2SOV385	Latency

Latency (Seconds) Average 5 Minutes Update Graph

Time Range: Relative Absolute UTC (GMT)

From: 3 days ago To: 0 minutes ago

Zoom: 1h | 3h | 6h | 12h | 1d | 3d | 1w | 2w

Cancel Back Next Create Alarm

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: `myHigh-CpuAlarm`, 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.

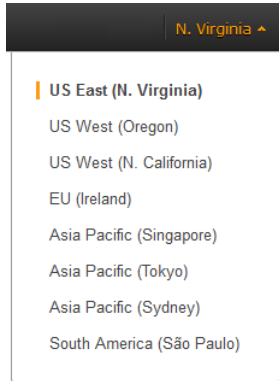
Send Email Based on Storage Throughput Alarm

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 Amazon 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 Volumeld.

A graph showing average `VolumeWriteBytes` for a single volume appears in the lower pane.

The screenshot shows the "Create Alarm" wizard in the AWS Management Console. The wizard is in the "1. Select Metric" step. The "Browse Metrics" pane shows a search for "EBS Metrics" and a list of metrics for the volume "vol-75192701". The "VolumeWriteBytes" metric is selected. Below the list is a graph showing the average "VolumeWriteBytes" for the selected volume over a 5-minute period. The graph shows a fluctuating line with a peak around 10,000 bytes. The "Time Range" section is set to "Relative" and "UTC (GMT)", with "From" set to "3 days ago" and "To" set to "0 minutes ago". The "Zoom" options are "1h", "3h", "6h", "12h", "1d", "3d", "1w", and "2w". The "Create Alarm" button is highlighted in blue.

Volumeld	Metric Name
vol-75192701	VolumeIdleTime
vol-75192701	VolumeQueueLength
vol-75192701	VolumeReadOps
vol-75192701	VolumeTotalWriteTime
vol-75192701	VolumeWriteBytes

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: `myHigh-CpuAlarm`, 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 or Terminate an Instance

Using Amazon CloudWatch alarm actions, you can create alarms that automatically stop or terminate your Amazon Elastic Compute Cloud (Amazon EC2) instances when you no longer need them to be running. 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 leave those instances sitting idle (and accruing 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 add the stop or terminate alarm actions to any alarm that is set on an Amazon EC2 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 Actions to Amazon CloudWatch Alarms \(p. 234\)](#)
- [Amazon CloudWatch Alarm Action Scenarios \(p. 244\)](#)

Adding Actions to Amazon CloudWatch Alarms

You can configure alarm actions using either the Amazon EC2 console or the Amazon CloudWatch console, or you can use the Amazon CloudWatch command line interface (CLI), 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

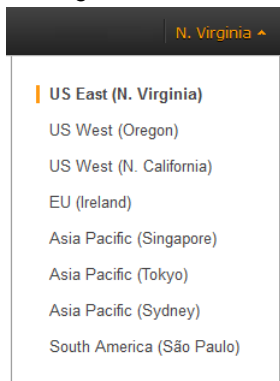
If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions: `ec2:DescribeInstanceStatus`, `ec2:DescribeInstances`, `ec2:StopInstances`, and `ec2:TerminateInstances` in order for the alarm action to be performed. 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 *Using IAM*.

If you are using an IAM role (e.g. 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.

To create an alarm to stop an idle instance

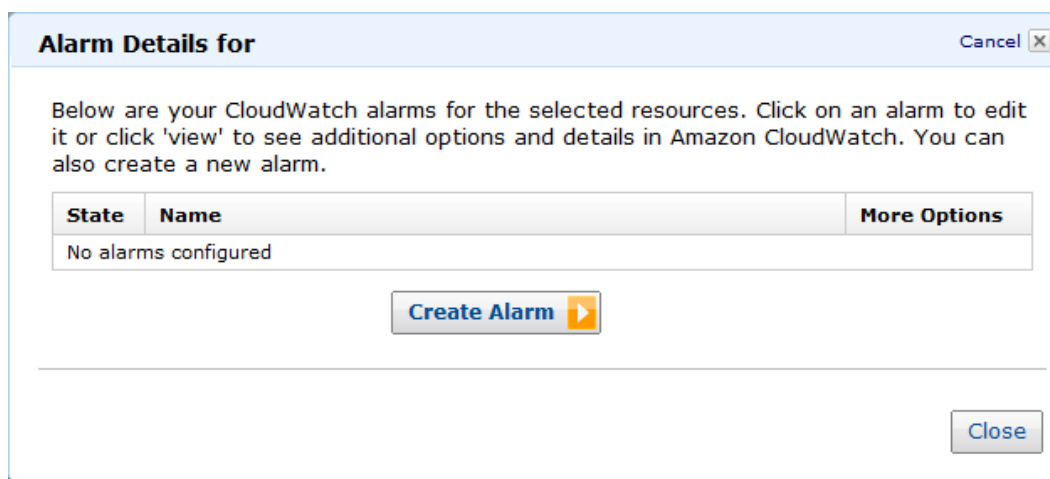
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see [Regions and Endpoints](#).



3. In the navigation pane, under **INSTANCES**, click **Instances**.
4. In the upper pane, right-click an instance, 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.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Create Alarms That Stop or Terminate an Instance**

Create Alarm for Cancel X

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define. To edit an alarm, first choose whom to notify and then define when the notification should be sent

Send a notification to: EC2InstanceStop cancel

With these recipients: john.stiles@example.com

Take the action: **Stop** Terminate this instance.

Whenever: Average of CPU Utilization

Is: < 10 Percent

For at least: 24 consecutive period(s) of 1 hour

Name of alarm: EC2InstanceStop

Cancel Create Alarm

7. Select the **Take the action** check box, and then choose the **Stop** radio button.
8. In the **Whenever** boxes, choose the statistic you want to use and then select the metric. In this example, choose **Average** and **CPU Utilization**.
9. In the **Is** boxes, define the metric threshold. In this example, enter **10** percent.
10. In the **For at least** box, choose the sampling period for the alarm. In this example, enter **24** consecutive periods of one hour.
11. 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 automatically creates 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.

12. Click **Create Alarm**.

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 about enabling and disabling termination protection for an instance, see [Enabling Termination Protection for an Instance](#) (p. 147).

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`, `ec2:DescribeInstances`, `ec2:StopInstances`, and `ec2:TerminateInstances` in order for the alarm action to be performed. 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 *Using IAM*.

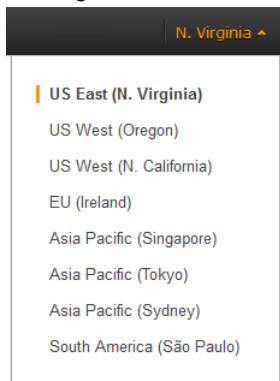
If you are using an IAM role (e.g. Amazon EC2 instance profile), you cannot stop or terminate the instance using alarm actions. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Create Alarms That Stop or Terminate an Instance**

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.

To create an alarm to terminate an idle instance

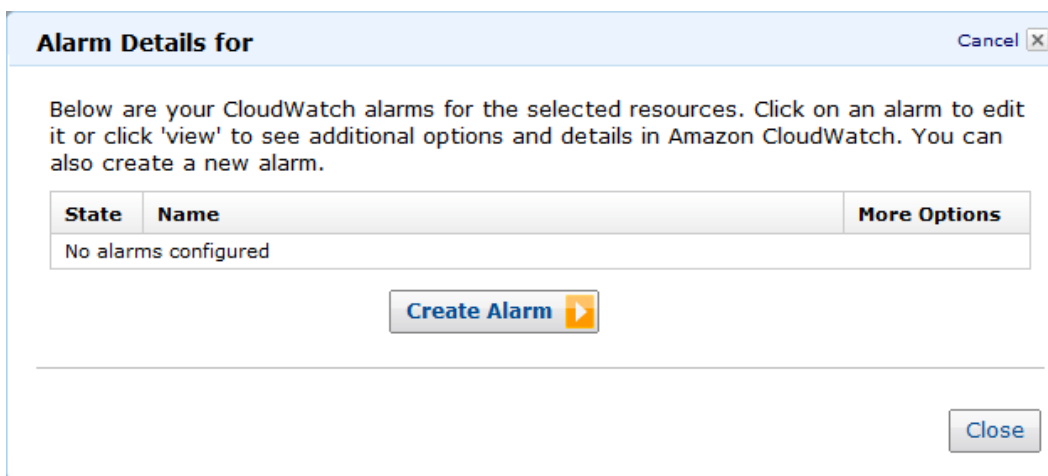
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see [Regions and Endpoints](#).



3. In the navigation pane, under **INSTANCES**, click **Instances**.
4. In the upper pane, right-click an instance, 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 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.

Amazon Elastic Compute Cloud User Guide for Microsoft Windows

Create Alarms That Stop or Terminate an Instance

Create Alarm for Cancel

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define. To edit an alarm, first choose whom to notify and then define when the notification should be sent

Send a notification to: EC2InstanceTerminate cancel

With these recipients: john.stiles@example.com

Take the action: Stop **Terminate** this instance.

Whenever: Average of CPU Utilization

Is: < 10 Percent

For at least: 24 consecutive period(s) of 1 hour

Name of alarm: EC2InstanceTerminate

Cancel Create Alarm

7. Select the **Take the action** check box, and then choose the **Terminate** radio button.
8. In the **Whenever** boxes, choose the statistic you want to use and then select the metric. In this example, choose **Average** and **CPU Utilization**.
9. In the **Is** boxes, define the metric threshold. In this example, enter **10** percent.
10. In the **For at least** box, choose the sampling period for the alarm. In this example, enter **24** consecutive periods of one hour.
11. 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 automatically creates 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.

12. 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`, `ec2:DescribeInstances`, `ec2:StopInstances`, and `ec2:TerminateInstances` in order for the alarm action to be performed. 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

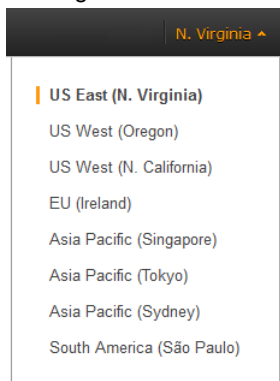
earlier will be performed. For more information about IAM permissions, see [Permissions and Policies](#) in *Using IAM*.

If you are using an IAM role (e.g. 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.

To create an alarm to stop an idle instance

1. Open the Amazon 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 the **CloudWatch Metrics by Category** pane, 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. 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: **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.

17. Click **Create Alarm** to complete the alarm creation process.

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 about disabling termination protection on an instance, see [Enabling Termination Protection for an Instance](#) (p. 147).

Note

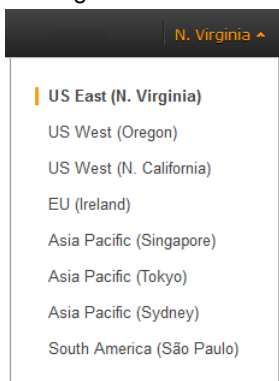
If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions: `ec2:DescribeInstanceStatus`, `ec2:DescribeInstances`, `ec2:StopInstances`, and `ec2:TerminateInstances` in order for the alarm action to be performed. 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 *Using IAM*.

If you are using an IAM role (e.g. 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.

To create an alarm to terminate an idle instance

1. Open the Amazon 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, click **Create Alarm**.
5. Click **Create Alarm**, and then in the **CloudWatch Metrics by Category** pane, under **EC2 Metrics**, select **Per-Instance Metrics**.

6. 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.
7. Select **Average** from the **Statistic** drop-down list.
8. Select a period from the **Period** drop-down list, for example: **1 Day**.
9. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: **Terminate EC2 instance**.
10. In the **Description** field, enter a description of the alarm, for example: **Terminate EC2 instance when CPU is idle for too long**.
11. In the **is** drop-down list, select **<**.
12. 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**.

13. Under **Actions**, click **EC2 Action**.
14. In the **Whenever this alarm** drop-down list, select **State is ALARM**.
15. In the **Take this action** drop-down list, select **Terminate this instance**.
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.

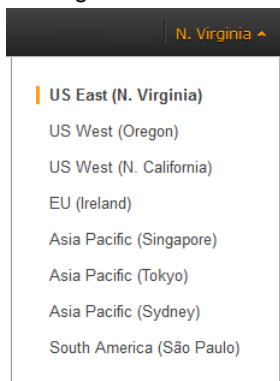
18. Click **Create Alarm** to complete the alarm creation process.

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 Amazon 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 or Terminate an Instance

If you are using either the AWS CLI or the CloudWatch API, or if you are using the AWS SDKs with the API, you can create an Amazon CloudWatch alarm using an Amazon EC2 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)
<i>Stop</i>	arn:aws:automate:us-east-1:ec2:stop
<i>Terminate</i>	arn:aws:automate:us-east-1:ec2:terminate

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`, `ec2:DescribeInstances`, `ec2:StopInstances`, and `ec2:TerminateInstances` in order for the alarm action to be performed. 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 *Using IAM*.

If you are using an IAM role (e.g. 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.

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 CPUUtilization 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 and to stop an instance using the API

The following example request shows how to create an alarm that stops an Amazon EC2 instance.

- Construct the following request:

```
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
&AWSSecretKeyId=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 and to terminate an instance using the API

The following example request shows how to create an alarm that terminates an Amazon EC2 instance.

- Construct the following request:

```
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  
  
&AWSSecretKeyId=XXX YOUR ACCESS KEY XXX  
  
&Signature=%XXX YOUR SIGNATURE XXX%3D  
  
&AlarmActions.member.1=arn:aws:automate:us-east-1:ec2:terminate
```

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

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Create Alarms That Stop or Terminate an Instance**

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.

Step	Value
1	Stop
2	Maximum
3	CPUUtilization
4	<=
5	10%
6	60 minutes
7	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.

Step	Value
1	Stop and email
2	Average
3	CPUUtilization
4	<=

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Create Alarms That Stop or Terminate an Instance**

Step	Value
5	5%
6	60 minutes
7	24

Scenario 3: Stop Web Servers with Unusually High Traffic

Create an alarm that sends email when an instance exceeds 10 GB of outbound network traffic per day.

Step	Value
1	Email
2	Sum
3	NetworkOut
4	>
5	10 GB
6	1 day
7	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.

Step	Value
1	Stop and send SMS
2	Sum
3	NetworkOut
4	>
5	1 GB
6	1 hour
7	1

Scenario 5: Stop an Instance Experiencing a Memory Leak

Create an alarm that stops an instance when memory utilization reaches or exceeds 90 percent, 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 [Monitoring Scripts for Amazon EC2 Instances](#) (p. 249).

Step	Value
1	Stop
2	Maximum
3	MemoryUtilization
4	>=
5	90%
6	1 minute
7	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).

Step	Value
1	Stop
2	Average
3	StatusCheckFailed_System
4	>=
5	1
6	15 minutes
7	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.

Amazon Elastic Compute Cloud User Guide for Microsoft Windows
Create Alarms That Stop or Terminate an Instance

Step	Value
1	Terminate
2	Maximum
3	NetworkOut
4	<=
5	100,000 bytes
6	5 minutes
7	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.

Create Alarm
✕

1. Select Metric
2. Define Alarm

Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name:

Description:

Whenever: CPUUtilization

is: 4 5

for: 7 consecutive period(s)

Actions

Define what actions are taken when your alarm changes state.

Notification Delete

Whenever this alarm: State is ALARM

1 Send notification to: Select a notification list [New list](#)

+ Notification
+ AutoScaling Action
+ EC2 Action

Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 5 minutes

Namespace: AWS/EC2

InstancedId: i-0c986c72

3 Metric Name: CPUUtilization

6 Period: 5 Minutes

2 Statistic: Average

Cancel
Back
Next
Create Alarm

Monitoring Scripts for Amazon EC2 Instances

The Amazon CloudWatch Monitoring Scripts for Windows instances demonstrate how to produce and consume Amazon CloudWatch custom metrics. The scripts for Windows are sample PowerShell scripts that comprise a fully functional example that reports memory, page file, and disk space utilization metrics for a Windows instance. You can download the CloudWatch Monitoring Scripts from the Amazon Web Services (AWS) sample code library and install them on your Windows instances.

Important

These scripts are examples only. They are provided "as is" and are not supported.

Note

Standard Amazon CloudWatch free tier quantities and usage charges for custom metrics apply to your use of these scripts. For more information, see the [Amazon CloudWatch](#) product page.

Amazon CloudWatch Monitoring Scripts for Windows

The Amazon CloudWatch Monitoring Scripts for Windows are sample PowerShell scripts that demonstrate how to produce and consume Amazon CloudWatch custom metrics. The scripts comprise a fully functional example that reports memory, page file, and disk space utilization metrics for an Amazon Elastic Compute Cloud (Amazon EC2) Windows instance.

These monitoring scripts are intended for use with Amazon EC2 instances running Microsoft Windows Server. The scripts have been tested on the following Amazon Machine Images (AMIs) for both 32-bit and 64-bit versions:

- Windows Server 2003 R2
- Windows Server 2008
- Windows Server 2008 R2

Contents

- [Getting Started](#) (p. 249)
- [mon-put-metrics-mem.ps1](#) (p. 250)
- [mon-put-metrics-disk.ps1](#) (p. 252)
- [mon-put-metrics-perfmon.ps1](#) (p. 254)
- [mon-get-instance-stats.ps1](#) (p. 256)
- [Set Up Task Scheduler to Send Metrics Reports to Amazon CloudWatch](#) (p. 257)

Getting Started

The following steps demonstrate how to download, uncompress, and configure the Amazon CloudWatch Monitoring Scripts on an Amazon EC2 Windows instance.

To download, install, and configure the script

1. Connect to your Amazon EC2 Windows instance. For information about how to connect to Amazon EC2 Windows instances, see [Connecting to Your Windows Instance Using RDP](#) (p. 138).
2. Download and install the [AWS SDK for .NET](#) onto the EC2 instance that you want to monitor.
3. Download the .zip file containing the [Amazon CloudWatch Monitoring Scripts for Microsoft Windows Server](#) onto the EC2 instance and unzip it in a location of your preference.

The `AmazonCloudWatchMonitoringWindows.zip` package contains these files:

- **mon-put-metrics-mem.ps1** —Collects system metrics on an Amazon EC2 Windows instance (memory, page file utilization) and sends them to Amazon CloudWatch.
- **mon-put-metrics-disk.ps1** —Collects system metrics on an Amazon EC2 instance (disk space utilization) and sends them to Amazon CloudWatch.
- **mon-put-metrics-perfmon.ps1** —Collects PerfMon counters on an Amazon EC2 instance and sends them to Amazon CloudWatch.
- **mon-get-instance-stats.ps1**—Queries Amazon CloudWatch and displays the most recent utilization statistics for the EC2 instance on which this script is executed.
- **awscreds.conf**—File template for AWS credentials that stores your access key ID and secret access key.
- **LICENSE.txt**—Text file containing the Apache 2.0 license.
- **NOTICE.txt**—Copyright notice.

4. Update the `awscreds.conf` file that you downloaded earlier. The content of this file should use the following format:

```
AWSAccessKeyId=YourAccessKeyID
```

```
AWSSecretKey=YourSecretAccessKey
```

Note

This step is optional if you have already created a file for credentials. You can use an existing file by specifying its location on the command line when you call the scripts. Alternatively, you can set the environment variable `AWS_CREDENTIAL_FILE` to point to the file with your AWS credentials.

For instructions on how to access your credentials, use the following procedure.

As a best practice, do not use the root credentials. Instead, you should create an Identity and Access Management (IAM) user with a policy that restricts the user to only Amazon CloudWatch operations. For more information, see [Controlling User Access to Your AWS Account](#).

mon-put-metrics-mem.ps1

This script collects memory and pagefile utilization data on the current system. It then makes a remote call to Amazon CloudWatch to report the collected data as custom metrics.

Options

Name	Description
<code>-mem_util</code>	Collects and sends the MemoryUtilization metrics in percentages. This option reports only memory allocated by applications and the operating system, and excludes memory in cache and buffers.
<code>-mem_used</code>	Collects and sends the MemoryUsed metrics, reported in megabytes. This option reports only memory allocated by applications and the operating system, and excludes memory in cache and buffers.
<code>-mem_avail</code>	Collects and sends the MemoryAvailable metrics, reported in megabytes. This option reports memory available for use by applications and the operating system.
<code>-page_util</code>	Collects and sends PageUtilization metrics, reported in percentages. Page utilization is reported for each page file in a windows instance.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Amazon CloudWatch Monitoring Scripts for Windows**

Name	Description
<code>-page_used</code>	Collects and sends PageUsed metrics, reported in megabytes.
<code>-page_avail</code>	Reports available space in page file for all disks.
<code>-memory_units UNITS</code>	Specifies units in which to report memory usage. If not specified, memory is reported in megabytes. UNITS may be one of the following: bytes, kilobytes, megabytes, gigabytes.
<code>-aws_credential_file=PATH</code>	Provides the location of the file containing AWS credentials. This parameter cannot be used with the <code>-aws_access_id</code> and <code>-aws_secret_key</code> parameters.
<code>-aws_access_id=VALUE</code>	Specifies the AWS access key ID to use to identify the caller. Must be used together with the <code>-aws_secret_key</code> option. Do not use this option with the <code>-aws_credential_file</code> option.
<code>-aws_secret_key=VALUE</code>	Specifies the AWS secret access key to use to sign the request to Amazon CloudWatch. Must be used together with the <code>-aws_access-key_id</code> option. Do not use this option with <code>-aws_credential_file</code> option.
<code>-whatif</code>	Performs a test run of the script that collects the metrics but does not actually call Amazon CloudWatch to report the data. This option also checks that credentials are provided.
<code>-from_scheduler</code>	Use this option when calling the script from task scheduler. When this option is used, all diagnostic output is suppressed, but error messages are sent to the log file.
<code>-verbose</code>	Displays detailed information about what the script is doing.
<code>Get-help</code> <code>mon-put-metrics-mem.ps1</code>	Displays usage information.
<code>-version</code>	Displays the version number of the script.
<code>-logfile</code>	Logfile is used to log error message. Use this along with <code>-from_scheduler</code> option. If no value is specified for logfile then a default file is created with the same as the script with <code>.log</code> extension.

Examples

The following examples assume that you have already updated the `awscreds.conf` file with valid AWS credentials. If you are not using the `awscreds.conf` file, provide credentials using the `-aws_access_id` and `-aws_secret_key` arguments.

To collect all available memory metrics using an inline access ID and secret key and send the data to CloudWatch

- Run the following command:

```
.\mon-put-metrics-mem.ps1 -aws_access_id ThisIsMyAccessKey -aws_secret_key
ThisIsMySecretKey -mem_util -mem_avail -page_avail -page_used -page_util
-memory_units Megabytes
```

To collect all available memory metrics using a credential file and send the data to CloudWatch

- Run the following command:

```
.\mon-put-metrics-mem.ps1 -aws_credential_file C:\awscreds.conf -mem_util
-mem_used -mem_avail -page_avail -page_used -page_util -memory_units Megabytes
```

To collect all available memory metrics using credentials stored in environment variables and send the data to CloudWatch

- Run the following command:

```
.\mon-put-metrics-mem.ps1 -mem_util -mem_used -mem_avail -page_avail -
page_used -page_util -memory_units Megabytes
```

mon-put-metrics-disk.ps1

This script collects disk space utilization data on the current system. It then makes a remote call to Amazon CloudWatch to report the collected data as custom metrics.

Options

Name	Description
<i>-disk_space_util</i>	Collects and sends the DiskSpaceUtilization metric for the selected disks. The metric is reported in percentages.
<i>-disk_space_used</i>	Collects and sends DiskSpaceUsed metric for the selected disks. The metric is reported by default in gigabytes.
<i>-disk_space_avail</i>	Collects and sends the DiskSpaceAvailable metric for the selected disks. The metric is reported in gigabytes.
<i>-disk_space_units</i> <i>UNITS</i>	Specifies units in which to report memory usage. If not specified, memory is reported in gigabytes. UNITS may be one of the following: bytes, kilobytes, megabytes, gigabytes.
<i>-disk_drive</i>	Selects the drive letter on which to report. To report metrics on the c and d drives, use the following option <i>-disk_drive C:, D:</i> Values should be comma separated.
<i>-aws_credential_file</i> <i>PATH</i>	Provides the location of the file containing AWS credentials. This parameter cannot be used with the <i>-aws_access_id</i> and <i>-aws_secret_key</i> parameters.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Amazon CloudWatch Monitoring Scripts for Windows**

Name	Description
<code>-aws_access_id VALUE</code>	Specifies the AWS access key ID to use to identify the caller. Must be used together with the <code>-aws_secret_key</code> option. Do not use this option with the <code>-aws_credential_file</code> option.
<code>-aws_secret_key VALUE</code>	Specifies the AWS secret access key to use to sign the request to Amazon CloudWatch. Must be used together with the <code>-aws_access_id</code> option. Do not use this option with <code>-aws_credential_file</code> option.
<code>-whatif</code>	Performs a test run of the script that collects the metrics but does not actually call Amazon CloudWatch to report the data. This option also checks that credentials are provided.
<code>-from_scheduler</code>	Use this option when calling the script from task scheduler. When this option is used, all diagnostic output is suppressed, but error messages are sent to the log file.
<code>-verbose</code>	Displays detailed information about what the script is doing.
<code>Get-help mon-put-metrics-disk.ps1</code>	Displays usage information.
<code>-version</code>	Displays the version number of the script.
<code>-logfile</code>	Logfile is used to log error message. Use this along with <code>-from_scheduler</code> option. If no value is specified for logfile then a default file is created with the same as the script with <code>.log</code> extension.

Examples

To collect all available disk metrics using an inline access ID and secret key and send the data to Amazon CloudWatch

- Run the following command:

```
.\mon-put-metrics-disk.ps1 -aws_access_id ThisIsMyAccessKey -aws_secret_key  
ThisIsMySecretKey -disk_space_util -disk_space_avail -disk_space_units  
Gigabytes
```

To collect all available disk metrics using a credential file and send the data to Amazon CloudWatch

- Run the following command:

```
.\mon-put-metrics-disk.ps1  
-aws_credential_file C:\awscreds.conf -disk_drive C:, D:  
-disk_space_util -disk_space_used -disk_space_avail -disk_space_units  
Gigabytes
```

To collect all available disk metrics using credentials stored in an environment variable and send the data to Amazon CloudWatch

- Run the following command:

```
.\mon-put-metrics-disk.ps1 -disk_drive C:, D:
                        -disk_space_util -disk_space_used -disk_space_avail -
disk_space_units Gigabytes
```

mon-put-metrics-perfmon.ps1

This script collects PerfMon counters on the current system. It then makes a remote call to Amazon CloudWatch to report the collected data as custom metrics.

Options

Name	Description
<i>-processor_queue</i>	Reports current processor queue counter.
<i>-pages_input</i>	Reports memory pages/input memory counter.
<i>-aws_credential_file</i> <i>PATH</i>	Provides the location of the file containing AWS credentials. This parameter cannot be used with the <i>-aws_access_id</i> and <i>-aws_secret_key</i> parameters.
<i>-aws_access_id</i> <i>VALUE</i>	Specifies the AWS access key ID to use to identify the caller. Must be used together with the <i>-aws_secret_key</i> option. Do not use this option with the <i>-aws_credential_file</i> option.
<i>-aws_secret_key</i> <i>VALUE</i>	Specifies the AWS secret access key to use to sign the request to Amazon CloudWatch. Must be used together with the <i>-aws_access_id</i> option. Do not use this option with <i>-aws_credential_file</i> option.
<i>-whatif</i>	Performs a test run of the script that collects the metrics but does not actually call Amazon CloudWatch to report the data. This option also checks that credentials are provided.
<i>-from_scheduler</i>	Use this option when calling the script from task scheduler. When this option is used, all diagnostic output is suppressed, but error messages are sent to the log file.
<i>-verbose</i>	Displays detailed information about what the script is doing.
<i>Get-help</i> <i>mon-put-metrics-disk.ps1</i>	Displays usage information.
<i>-version</i>	Displays the version number of the script.
<i>-logfile</i>	Logfile is used to log error message. Use this along with <i>-from_scheduler</i> option. If no value is specified for logfile then a default file is created with the same as the script with .log extension.

Examples

To collect preset PerfMon counters in script using an inline access ID and secret key and send the data to Amazon CloudWatch

- Run the following command:

```
.\mon-put-metrics-perfmon.ps1 -aws_access_id ThisIsMyAccessKey -aws_secret_key  
ThisIsMySecretKey -pages_input -processor_queue
```

To collect preset PerfMon counters in script using a credential file and send the data to Amazon CloudWatch

- Run the following command:

```
.\mon-put-metrics-perfmon.ps1 -aws_credential_file C:\awscreds.conf -  
pages_input -processor_queue
```

To collect preset PerfMon counters in script using credentials stored in an environment variable and send the data to Amazon CloudWatch

- Run the following command:

```
.\mon-put-metrics-perfmon.ps1 -pages_input -processor_queue
```

To add more counters to be pushed to Amazon CloudWatch

- Open the script in a text editor such as Notepad, and then on line 72, locate the following commented section:

```
### Add More counters here.  
#$Counters.Add('\Memory\Cache Bytes','Bytes')  
#$Counters.Add('\localhost\physicaldisk(0 c:)\% disk time','Percent')
```

Note

The first parameter (e.g., `$Counters.Add`) is the PerfMon counter. The second parameter (e.g., `(\Memory\Cache Bytes,'Bytes')`) is the unit of data that counter provides.

- Edit the script and add your own PerfMon counters to the script as shown above. After you have added custom PerfMon counters to the script, you can run the script without any parameters other than credential information.

Note

You can only add PerfMon counters to the script on your computer. You can use the `Get-Counter` command to test PerfMon counters. For more information, see [Get-Counter](#) on the Microsoft TechNet website.

mon-get-instance-stats.ps1

This script queries Amazon CloudWatch for statistics on memory, page file, and disk space metrics within the time interval provided using the number of most recent hours. This data is provided for the Amazon EC2 instance on which this script is executed.

Options

Name	Description
<code>-recent-hours N</code>	Specifies the number of recent hours to report on, as represented by N where N is an integer.
<code>-aws_credential_file PATH</code>	Provides the location of the file containing AWS credentials. This parameter cannot be used with the <code>-aws_access_id</code> and <code>-aws_secret_key</code> parameters.
<code>-aws_access_id VALUE</code>	Specifies the AWS access key ID to use to identify the caller. Must be used together with the <code>-aws_secret_key</code> option. Do not use this option with the <code>-aws_credential_file</code> option.
<code>-aws_secret_key VALUE</code>	Specifies the AWS secret access key to use to sign the request to Amazon CloudWatch. Must be used together with the <code>-aws_access_id</code> option. Do not use this option with <code>-aws_credential_file</code> option.
<code>-verbose</code>	Displays detailed information about what the script is doing.
<code>Get-help mon-get-instance-stats.ps1</code>	Displays usage information.
<code>-version</code>	Displays the version number of the script.

Examples

To get utilization statistics for the last 12 hours using an inline access ID and secret key and send the data to Amazon CloudWatch

- Run the following command:

```
.\ mon-get-instance-stats.ps1 -aws_access_id  
    ThisIsMyAccessKey -aws_secret_key ThisIsMySecretKey -re  
cent_hours 12
```

To get utilization statistics for the last 12 hours using a credential file and send the data to Amazon CloudWatch

- Run the following command:

```
.\mon-get-instance-stats.ps1 -aws_credential_file C:\awscreds.conf -re  
cent_hours 12
```

To get utilization statistics for the last 12 hours using credentials stored in an environment variable and send the data to Amazon CloudWatch

- Run the following command:

```
.\mon-get-instance-stats.ps1 -recent_hours 12
```

The returned response will be similar to the following example output:

```
Assembly Loaded
Instance Metrics for last 12 hours.
CPU Utilization
Average: 4.69 % Maximum: 10.47 % Minimum: 1.16 %

Memory Utilization
Average: 14.45 % Maximum: 14.77 % Minimum: 14.38 %

pagefileUtilization(c:\pagefile.sys)
Average: 0.00 % Maximum: 0.00 % Minimum: 0.00 %

Volume Utilization C:
Average: 17.28 % Maximum: 17.28 % Minimum: 17.28 %

Volume Utilization D:
Average: 1.41 % Maximum: 1.41 % Minimum: 1.41 %

pagefileUtilization(f:\pagefile.sys)
Average: 0.00 % Maximum: 0.00 % Minimum: 0.00 %
pagefileUtilization(f:\pagefile.sys)
Average: 0 Maximum: 0 Minimum: 0

pagefileUtilization(f:\pagefile.sys)
Average: 0 Maximum: 0 Minimum: 0
```

Set Up Task Scheduler to Send Metrics Reports to Amazon CloudWatch

You can use Windows Task Scheduler to send metrics reports periodically to Amazon CloudWatch.

To set up task scheduler to send metrics reports to Amazon CloudWatch

1. On your Windows Server instance, click **Start**, click **Administrative Tools**, and then click **Task Scheduler**.
2. On the **Action** menu, click **Create Task**.
3. In the **Create Task** dialog box, on the **General** tab, in the **Name** box, type a name for the task, and then select **Run whether user is logged on or not**.
4. On the **Triggers** tab, click **New**.
5. In the **New Trigger** dialog box, under **Settings**, select **One time**.
6. Under **Advanced settings**, select **Repeat task every** and select **5 minutes** from the drop-down menu.
7. In the **for a duration of** drop-down menu, select **Indefinitely**, and then click **OK**.

Note

These settings create a trigger that will launch the script every 5 minutes indefinitely. To modify this task to run for set number of days using the **Expire** check box.

8. On the **Actions** tab, click **New**.
9. In the **Action** drop-down menu, select **Start a program**.
10. Under **Settings**, in the **Program/script** box, type **Powershell.exe**.
11. In the **Add arguments (optional)** box, type `-command "C:\scripts\mon-put-metrics-disk.ps1 -disk_drive C:,d -disk_space_util -disk_space_units gigabytes -from_scheduler -logfile C:\mylogfile.log"`, and then click **OK**.
12. On the **Create Task** dialog box, click **OK**.

If you selected a user account to run this task, Task Scheduler will prompt you for user credentials. Enter the user name and password for the account that will run the task, and then click **OK**.

Note

If the PerfMon counters you are using don't require administrator privileges, you can run this task using a system account instead of an administrator account. In the **Create Task** dialog box, on the **General** tab, click **Change User or Group**, and then select a system account.

Network and Security

Amazon EC2 provides the following network and security features.

Features

- [Amazon EC2 Key Pairs](#) (p. 260)
- [Amazon EC2 Security Groups](#) (p. 264)
- [Controlling Access to Amazon EC2 Resources](#) (p. 272)
- [Amazon EC2 and Amazon Virtual Private Cloud \(VPC\)](#) (p. 309)
- [Amazon EC2 Instance IP Addressing](#) (p. 320)
- [Elastic IP Addresses \(EIP\)](#) (p. 329)
- [Elastic Network Interfaces \(ENI\)](#) (p. 334)
- [Enabling Enhanced Networking on Windows Instances in a VPC](#) (p. 346)

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. 312).

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

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.

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 \(p. 260\)](#).

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 \(p. 261\)](#).

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 1024-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 your private key, there is no way to recover it. If you lose the private key for an instance store-backed instance, you can't access the instance; you should terminate the instance and launch another instance using a new key pair.

Topics

- [Creating Your Key Pair Using Amazon EC2 \(p. 260\)](#)
- [Importing Your Own Key Pair to Amazon EC2 \(p. 261\)](#)
- [Retrieving the Public Key for Your Key Pair \(p. 263\)](#)
- [Verifying Your Key Pair's Fingerprint \(p. 263\)](#)
- [Deleting Your Key Pair \(p. 264\)](#)

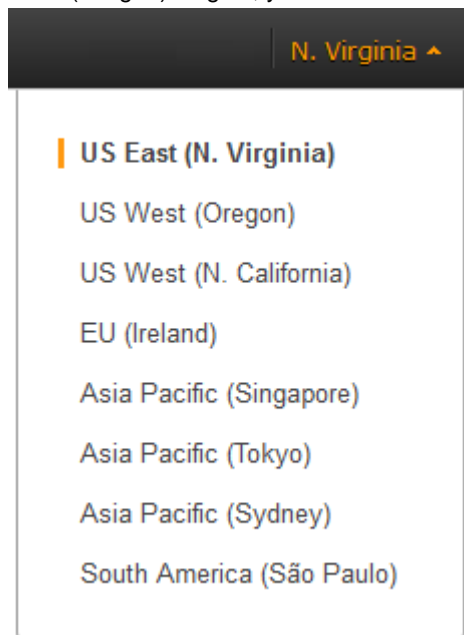
Creating Your Key Pair Using Amazon EC2

You can create a key pair using the Amazon EC2 console or the command line.

To create your key pair using the console

1. 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. 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. 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**.
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.

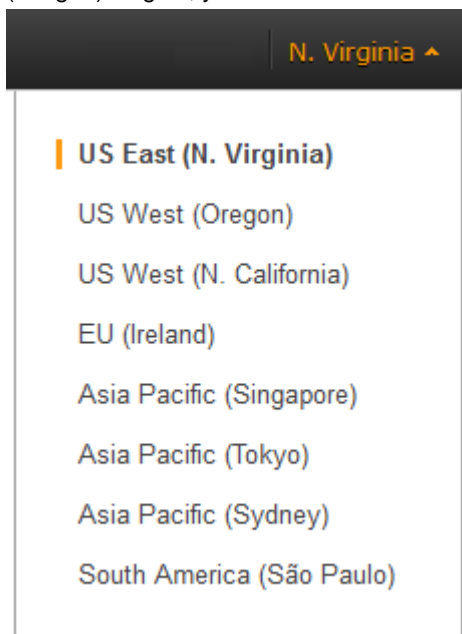
To create a key pair using a third-party tool

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.
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. Click **Key Pairs** in the navigation pane.
4. Click **Import Key Pair**.
5. In the **Import Key Pair** dialog box, click **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 click **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. From the navigation bar, select the region in which you created the key pair.
2. Click **Key Pairs** in the navigation pane.
3. 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, 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 AAAAB3NzaC1yc2EAAAADAQABAAQAClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtXJMXLvvwJryDUilBMTjYtwB+QhYXUMozce5Pjz5/i8SeJtjnv3iAoG/cQk+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwi3f05p6KLxEXAMPLE my-key-pair
```

For more information, see [Retrieving Instance Metadata](#) (p. 100).

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.
2. Click **Key Pairs** in the navigation pane.
3. Select the key pair and click **Delete**.
4. When prompted, click **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

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.

Topics

- [Security Groups for EC2-Classic \(p. 265\)](#)
- [Security Groups for EC2-VPC \(p. 265\)](#)
- [Security Group Rules \(p. 265\)](#)
- [Default Security Groups \(p. 266\)](#)
- [Custom Security Groups \(p. 267\)](#)
- [Creating a Security Group \(p. 268\)](#)
- [Describing Your Security Groups \(p. 268\)](#)
- [Adding Rules to a Security Group \(p. 269\)](#)
- [Deleting Rules from a Security Group \(p. 270\)](#)
- [Deleting a Security Group \(p. 270\)](#)
- [API and Command Overview \(p. 270\)](#)

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. You can also 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 Virtual Private Cloud 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 \(p. 320\)](#).

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.

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 Virtual Private Cloud 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-xxxxxxx. 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 \(p. 269\)](#).

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 Virtual Private Cloud 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 **EC2 security groups** from the filter list.
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 **VPC security groups** from the filter list.

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.

Type	Protocol	Port Range	Source
HTTP	TCP	80	Anywhere 0.0.0.0/0
HTTPS	TCP	443	Anywhere 0.0.0.0/0

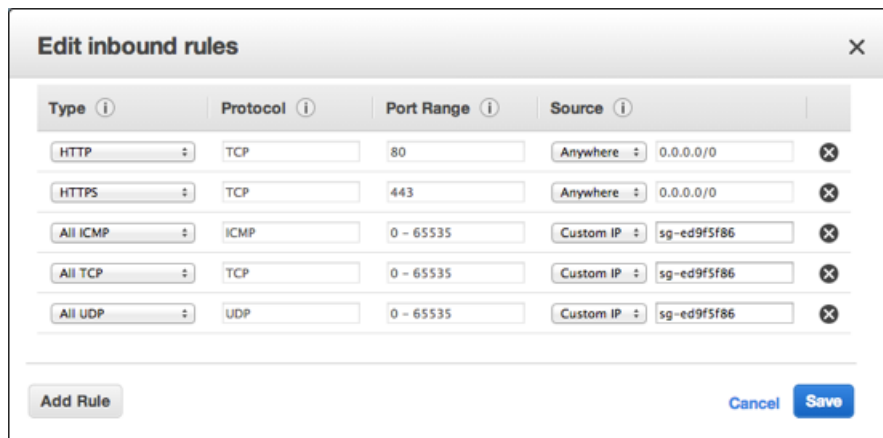
5. To connect to a Windows instance, you need to allow RDP traffic. Click **Add Rule**, and then select **RDP** from the **Type** list.

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 Virtual Private Cloud 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 associated with an instance. You can't delete the default 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 choose to allow full use or limited use of your Amazon EC2 resources.

Topics

- [Network Access to Your Instance](#) (p. 272)
- [Amazon EC2 Permission Attributes](#) (p. 272)
- [IAM and Amazon EC2](#) (p. 272)
- [IAM Policies for Amazon EC2](#) (p. 274)
- [IAM Roles for Amazon EC2](#) (p. 302)
- [Authorizing Inbound Traffic for Your Instances](#) (p. 308)

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 Instances](#) (p. 308).

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. 55).

Each Amazon EBS snapshot has a `createVolumePermission` attribute that controls which AWS accounts can use the snapshot. For more information, see [Sharing Snapshots](#) (p. 386).

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. In the **Select Policy Template** section, click **Select** next to a policy template of your choice. For example, for Amazon EC2, one of the following policy templates might meet your needs:
 - Power User Access
 - Read Only Access
 - Amazon EC2 Full Access
 - Amazon EC2 Read Only Access
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. 274)
- [IAM Roles for Amazon EC2](#) (p. 302)
- [Identity and Access Management \(IAM\)](#)
- [Using IAM](#)

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 *Using IAM* guide.

Topics

- [Policy Syntax](#) (p. 274)
- [Actions for Amazon EC2](#) (p. 275)
- [Amazon Resource Names for Amazon EC2](#) (p. 275)
- [Condition Keys for Amazon EC2](#) (p. 278)
- [Checking that Users Have the Required Permissions](#) (p. 280)
- [Supported Resources and Conditions for Amazon EC2 API Actions](#) (p. 281)
- [Example Policies for Working With the AWS CLI, the Amazon EC2 CLI, or an AWS SDK](#) (p. 287)
- [Example Policies for Working in the Amazon EC2 Console](#) (p. 294)

Policy Syntax

An IAM policy is a JSON document that consists of one or 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 \(p. 275\)](#).
- **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 \(p. 275\)](#). For more information about which API actions support which ARNs, see [Supported Resources and Conditions for Amazon EC2 API Actions \(p. 281\)](#). 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 \(p. 278\)](#).

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 \(p. 287\)](#).

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 Elastic Compute Cloud 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 Resources and Conditions for Amazon EC2 API Actions \(p. 281\)](#).

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	<code>arn:aws:ec2:*</code>
All Amazon EC2 resources owned by the specified account in the specified region	<code>arn:aws:ec2:region:account:*</code>
Customer gateway	<code>arn:aws:ec2:region:account:customer-gateway/cgw-id</code> Where <i>cgw-id</i> is <code>cgw-xxxxxxx</code>
DHCP options set	<code>arn:aws:ec2:region:account:dhcp-options/dhcp-options-id</code> Where <i>dhcp-options-id</i> is <code>dopt-xxxxxxx</code>
Image	<code>arn:aws:ec2:region:image/image-id</code> Where <i>image-id</i> is the ID of the AMI, AKI, or ARI, and <i>account</i> isn't used
Instance	<code>arn:aws:ec2:region:account:instance/instance-id</code> Where <i>instance-id</i> is <code>i-xxxxxxx</code>
Instance profile	<code>arn:aws:iam::account:instance-profile/instance-profile-name</code> Where <i>instance-profile-name</i> is the name of the instance profile, and <i>region</i> isn't used

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
IAM Policies**

Resource Type	ARN
Internet gateway	<code>arn:aws:ec2:region:account:internet-gateway/igw-id</code> Where <i>igw-id</i> is <code>igw-xxxxxxx</code>
Key pair	<code>arn:aws:ec2:region:account:key-pair/key-pair-name</code> Where <i>key-pair-name</i> is the key pair name (for example, <code>gsg-keypair</code>)
Network ACL	<code>arn:aws:ec2:region:account:network-acl/nacl-id</code> Where <i>nacl-id</i> is <code>acl-xxxxxxx</code>
Network interface	<code>arn:aws:ec2:region:account:network-interface/eni-id</code> Where <i>eni-id</i> is <code>eni-xxxxxxx</code>
Placement group	<code>arn:aws:ec2:region:account:placement-group/placement-group-name</code> Where <i>placement-group-name</i> is the placement group name (for example, <code>my-cluster</code>)
Route table	<code>arn:aws:ec2:region:account:route-table/route-table-id</code> Where <i>route-table-id</i> is <code>rtb-xxxxxxx</code>
Security group	<code>arn:aws:ec2:region:account:security-group/security-group-id</code> Where <i>security-group-id</i> is <code>sg-xxxxxxx</code>
Snapshot	<code>arn:aws:ec2:region::snapshot/snapshot-id</code> Where <i>snapshot-id</i> is <code>snap-xxxxxxx</code> , and <i>account</i> isn't used
Subnet	<code>arn:aws:ec2:region:account:subnet/subnet-id</code> Where <i>subnet-id</i> is <code>subnet-xxxxxxx</code>
Volume	<code>arn:aws:ec2:region:account:volume/volume-id</code> Where <i>volume-id</i> is <code>vol-xxxxxxx</code>
VPC	<code>arn:aws:ec2:region:account:vpc/vpc-id</code> Where <i>vpc-id</i> is <code>vpc-xxxxxxx</code>
VPC peering connection	<code>arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection-id</code> Where <i>vpc-peering connection-id</i> is <code>pcx-xxxxxxx</code>

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 Elastic Compute Cloud 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 *Using IAM* 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": " <i>vpc-arn</i> " Where <i>vpc-arn</i> is the VPC ARN for the peer VPC	ARN, Null
ec2:AvailabilityZone	"ec2:AvailabilityZone": " <i>az-api-name</i> " Where <i>az-api-name</i> is the name of the Availability Zone (for example, <i>us-west-2a</i>) To list your Availability Zones, use ec2-describe-availability-zones	String, Null
ec2:EbsOptimized	"ec2:EbsOptimized": " <i>optimized-flag</i> " Where <i>optimized-flag</i> is <code>true</code> <code>false</code>	Boolean, Null
ec2:ImageType	"ec2:ImageType": " <i>image-type-api-name</i> " Where <i>image-type-api-name</i> is <code>ami</code> <code>aki</code> <code>ari</code>	String, Null
ec2:InstanceProfile	"ec2:InstanceProfile": " <i>instance-profile-arn</i> " Where <i>instance-profile-arn</i> is the instance profile ARN	ARN, Null

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
IAM Policies**

Condition Key	Key/Value Pair	Evaluation Types
ec2:InstanceType	"ec2:InstanceType": <i>"instance-type-api-name"</i> Where <i>instance-type-api-name</i> is the name of the instance type (t2.micro t2.small t2.medium m3.medium m3.large m3.xlarge m3.2xlarge m1.small m1.medium m1.large m1.xlarge 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 hs1.8xlarge hi1.4xlarge t1.micro g2.2xlarge cg1.4xlarge).	String, Null
ec2:Owner	"ec2:Owner": <i>"account-id"</i> Where <i>account-id</i> is amazon <i>aws-account-id</i>	String, Null
ec2:ParentSnapshot	"ec2:ParentSnapshot": <i>"snapshot-arn"</i> Where <i>snapshot-arn</i> is the snapshot ARN	ARN, Null
ec2:ParentVolume	"ec2:ParentVolume": <i>"volume-arn"</i> Where <i>volume-arn</i> is the volume ARN	ARN, Null
ec2:PlacementGroup	"ec2:PlacementGroup": <i>"placement-group-arn"</i> Where <i>placement-group-arn</i> is the placement group ARN	ARN, Null
ec2:PlacementGroupStrategy	"ec2:PlacementGroupStrategy": <i>"placement-group-strategy"</i> Where <i>placement-group-strategy</i> is cluster	String, Null
ec2:Public	"ec2:Public": <i>"public-flag"</i> Where <i>public-flag</i> for an AMI is true false	Boolean, Null
ec2:Region	"ec2:Region": <i>"region-name"</i> Where <i>region-name</i> 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": <i>"vpc-arn"</i> Where <i>vpc-arn</i> is the VPC ARN for the requester's VPC	ARN, Null
ec2:ResourceTag/tag-key	"ec2:ResourceTag/tag-key": <i>"tag-value"</i> Where <i>tag-key</i> and <i>tag-value</i> are the tag-key pair	String, Null
ec2:RootDeviceType	"ec2:RootDeviceType": <i>"root-device-type-name"</i> Where <i>root-device-type-name</i> is ebs instance-store	String, Null
ec2:Subnet	"ec2:Subnet": <i>"subnet-arn"</i> Where <i>subnet-arn</i> is the subnet ARN	ARN, Null

Condition Key	Key/Value Pair	Evaluation Types
ec2:Tenancy	"ec2:Tenancy": <i>tenancy-attribute</i> Where <i>tenancy-attribute</i> is <code>default</code> <code>dedicated</code>	String, Null
ec2:Volumelops	"ec2:Volumelops": <i>volume-iops</i> Where <i>volume-iops</i> is the input/output operations per second (IOPS); the range is 100 to 4,000	Numeric, Null
ec2:VolumeSize	"ec2:VolumeSize": <i>volume-size</i> Where <i>volume-size</i> is the size of the volume, in GiB	Numeric, Null
ec2:VolumeType	"ec2:VolumeType": <i>volume-type-name</i> Where <i>volume-type-name</i> is <code>gp2</code> for General Purpose (SSD) volumes, <code>standard</code> for Magnetic Amazon EBS volumes, or <code>io1</code> for Provisioned IOPS (SSD) volumes.	String, Null
ec2:Vpc	"ec2:Vpc": <i>vpc-arn</i> Where <i>vpc-arn</i> 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 Resources and Conditions for Amazon EC2 API Actions \(p. 281\)](#). For example policy statements for Amazon EC2, see [Example Policies for Working With the AWS CLI, the Amazon EC2 CLI, or an AWS SDK \(p. 287\)](#).

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 Resources and Conditions for Amazon EC2 API Actions

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. We'll add support for additional actions, ARNs, and condition keys later.

API Action	Resources	Condition Keys
AcceptVpcPeeringConnection	VPC peering connection <i>arn:aws:ec2:region:account:peeringconnection/vpc-peering-connection-id</i>	ec2:AccepterVpc ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RequesterVpc
	VPC <i>arn:aws:ec2:region:account:vpc/vpc-id</i> Where <i>vpc-id</i> is a VPC owned by the accepter.	ec2:ResourceTag/ <i>tag-key</i> ec2:Region ec2:Tenancy
AttachVolume	Instance <i>arn:aws:ec2:region:account:instance/instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType ec2:Tenancy
	Volume <i>arn:aws:ec2:region:account:volume/volume-id</i>	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:VolumeIops ec2:VolumeSize ec2:VolumeType

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
IAM Policies**

API Action	Resources	Condition Keys
AuthorizeSecurityGroupEgress	Security group <i>arn:aws:ec2:region:account:security-group/security-group-id</i>	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
AuthorizeSecurityGroupIngress	Security group <i>arn:aws:ec2:region:account:security-group/security-group-id</i>	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
CreateVpcPeeringConnection	VPC <i>arn:aws:ec2:region:account:vpc/vpc-id</i> Where <i>vpc-id</i> is a requester VPC.	ec2:ResourceTag/tag-key ec2:Region ec2:Tenancy
	VPC peering connection <i>arn:aws:ec2:region:account:vpc-peering-connection*</i>	ec2:AccepterVpc ec2:Region ec2:RequesterVpc
DeleteCustomerGateway	Customer gateway <i>arn:aws:ec2:region:account:customer-gateway/ogw-id</i>	ec2:Region ec2:ResourceTag/tag-key
DeleteDhcpOptions	DHCP options set <i>arn:aws:ec2:region:account:dhcp-options/dhcp-options-id</i>	ec2:Region ec2:ResourceTag/tag-key
DeleteInternetGateway	Internet gateway <i>arn:aws:ec2:region:account:internet-gateway/igw-id</i>	ec2:Region ec2:ResourceTag/tag-key
DeleteNetworkAcl	Network ACL <i>arn:aws:ec2:region:account:network-acl/nac-id</i>	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
DeleteNetworkAclEntry	Network ACL <i>arn:aws:ec2:region:account:network-acl/nac-id</i>	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
DeleteRoute	Route table <i>arn:aws:ec2:region:account:route-table/route-table-id</i>	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
DeleteRouteTable	Route table <i>arn:aws:ec2:region:account:route-table/route-table-id</i>	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
IAM Policies**

API Action	Resources	Condition Keys
DeleteSecurityGroup	Security group <i>arn:aws:ec2:region:account:security-group/security-group-id</i>	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
DeleteVolume	Volume <i>arn:aws:ec2:region:account:volume/volume-id</i>	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:ResourceTag/tag-key ec2:Volumelops ec2:VolumeSize ec2:VolumeType
DeleteVpcPeeringConnection	VPC peering connection <i>arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection-id</i>	ec2:AccepterVpc ec2:Region ec2:ResourceTag/tag-key ec2:RequesterVpc
DetachVolume	Instance <i>arn:aws:ec2:region:account:instance/instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
	Volume <i>arn:aws:ec2:region:account:volume/volume-id</i>	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:ResourceTag/tag-key ec2:Volumelops ec2:VolumeSize ec2:VolumeType

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
IAM Policies**

API Action	Resources	Condition Keys
RebootInstances	Instance <i>arn:aws:ec2:region:account:instance/instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType ec2:Tenancy
RejectVpcPeeringConnection	VPC peering connection <i>arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection-id</i>	ec2:AcceptorVpc ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RequesterVpc
RevokeSecurityGroupEgress	Security group <i>arn:aws:ec2:region:account:security-group/security-group-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
RevokeSecurityGroupIngress	Security group <i>arn:aws:ec2:region:account:security-group/security-group-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
IAM Policies**

API Action	Resources	Condition Keys
RunInstances	Image <i>arn:aws:ec2:region::image/image-id</i>	ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/ <i>tag-key</i>
	Instance <i>arn:aws:ec2:region:accountinstance/instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	Key pair <i>arn:aws:ec2:region:accountkey-pair/key-pair-name</i>	ec2:Region
	Network interface <i>arn:aws:ec2:region:accountnetwork-interface/*</i> <i>arn:aws:ec2:region:accountnetwork-interface/eni-id</i>	ec2:AvailabilityZone ec2:Region ec2:Subnet ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
	Placement group <i>arn:aws:ec2:region:accountplacementgroup/placementgroup-name</i>	ec2:Region ec2:PlacementGroupStrategy
	Security group <i>arn:aws:ec2:region:accountsecurity-group/security-group-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
	Snapshot <i>arn:aws:ec2:region::snapshot/snapshot-id</i>	

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
IAM Policies**

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 <i>arn:aws:ec2:region:account:instance/instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType ec2:Tenancy
TerminateInstances	Instance <i>arn:aws:ec2:region:account:instance/instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> 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. 294). 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. 287)
- 2: Allow users to describe, launch, stop, start, and terminate all instances (p. 288)
- 3: Allow users to describe all instances, and stop, start, and terminate only particular instances (p. 288)
- 4: Allow users to manage particular volumes for particular instances (p. 289)
- 5: Allow users to launch instances with a specific configuration (p. 290)

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 Resources and Conditions for Amazon EC2 API Actions \(p. 281\)](#).

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 Resources and Conditions for Amazon EC2 API Actions \(p. 281\)](#).

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 (Northern 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 Resources and Conditions for Amazon EC2 API Actions \(p. 281\)](#).

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 (Northern 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 \(p. 288\)](#).

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 `t1.micro` or `m1.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 `t1.micro` or `m1.small`.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region:account:instance*"
    ],
    "Condition": {
      "StringEquals": {
        "ec2:InstanceType": ["t1.micro", "m1.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*"
    ]
  }
]
```

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:account: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:account: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 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 \(p. 295\)](#)
- [2: Using the EC2 launch wizard \(p. 296\)](#)
- [3: Working with volumes \(p. 299\)](#)
- [4: Working with security groups \(p. 300\)](#)
- [5: Working with Elastic IP addresses \(p. 302\)](#)

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 \(p. 287\)](#). Users cannot perform any actions on those resources or create new resources, unless another statement grants them permission to do so.

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; 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 Resources and Conditions for Amazon EC2 API Actions \(p. 281\)](#).

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.

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.

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 \(p. 290\)](#). 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.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
IAM Policies**

```
{
  "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

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 \(p. 264\)](#).

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"
      }
    }
  }
]
```

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-Classical 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 *Using IAM*.

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 *Using IAM* guide.

Topics

- [Retrieving Security Credentials from Instance Metadata](#) (p. 303)
- [Granting an IAM User Permission to Launch an Instance with an IAM Role](#) (p. 304)
- [Launching an Instance with an IAM Role Using the Console](#) (p. 304)
- [Launching an Instance with an IAM Role Using the AWS CLI](#) (p. 305)
- [Launching an Instance with an IAM Role Using an AWS SDK](#) (p. 307)

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 \(p. 100\)](#). 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 *Using IAM* guide.

Launching an Instance with an IAM Role Using the Console

You must create an IAM role before you can launch an instance with that role.

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 *Using IAM* guide.

To create an IAM role using the IAM console

1. Open the IAM console.
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 **Set Permissions** page, specify the policies for the group. You can select a policy template or create custom policies. For example, for Amazon EC2, one of the following policy templates might meet your needs:
 - Power User Access
 - Read Only Access

- Amazon EC2 Full Access
- Amazon EC2 Read Only Access

For more information about creating custom policies, see [IAM Policies for Amazon EC2 \(p. 274\)](#).

6. On the second **Set Permissions** page, you can replace the automatically generated policy name with a name of your choice. Check the details in the policy document, and click **Next Step**.
7. Review the role information, edit the role as needed, and then click **Create Role**.

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
```

Launching an Instance with an IAM Role Using the AWS CLI

You must create an IAM role before you can launch an instance with that role.

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 *Using IAM* guide.

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-document 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-Permissions --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-Permissions
{
  "InstanceProfile": {
    "InstanceId": "AIPAJTLPJLEGREXAMPLE",
    "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*.

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-credentials/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. The AWS SDK documentation includes walkthroughs that show how an application can use security credentials from a IAM role to read an Amazon S3 bucket. 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 Instances

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.

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 \(p. 264\)](#).

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 \(p. 308\)](#).

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 Virtual Private Cloud User Guide*.

Amazon EC2 and Amazon Virtual Private Cloud (VPC)

Amazon Virtual Private Cloud (Amazon VPC) enables you to define a virtual network in your own logically isolated area within the Amazon Web Services (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 datacenter, 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 datacenter. 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 Virtual Private Cloud User Guide](#).

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-Classical and EC2-VPC

Instances run in one of two supported platforms: EC2-Classical and EC2-VPC. Your AWS account is capable of launching instances either into both platforms or only into EC2-VPC, on a region by region basis. If you can launch instances only into EC2-VPC, we create a default VPC for you. A default VPC combines the benefits of the advanced features provided by EC2-VPC with the ease of use of EC2-Classical. For more information, see [Supported Platforms \(p. 312\)](#).

The following table summarizes the differences between instances launched in EC2-Classical, instances launched in a default VPC, and instances launched in a nondefault VPC.

Characteristic	EC2-Classical	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-Classical 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.
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.

Characteristic	EC2-Classical	Default VPC	Nondefault VPC
Security group association	<p>You can assign an unlimited number of security groups to an instance when you launch it.</p> <p>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).</p>	<p>You can assign up to 5 security groups to an instance.</p> <p>You can assign security groups to your instance when you launch it and while it's running.</p>	<p>You can assign up to 5 security groups to an instance.</p> <p>You can assign security groups to your instance when you launch it and while it's running.</p>
Security group rules	<p>You can add rules for inbound traffic only.</p> <p>You can add up to 100 rules to a security group.</p>	<p>You can add rules for inbound and outbound traffic.</p> <p>You can add up to 50 rules to a security group.</p>	<p>You can add rules for inbound and outbound traffic.</p> <p>You can add up to 50 rules to a security group.</p>
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.

Amazon VPC Documentation

For more information about Amazon VPC, see the Amazon VPC documentation.

Guide	Description
Amazon Virtual Private Cloud Getting Started Guide	Provides a hands-on introduction to Amazon VPC.
Amazon Virtual Private Cloud User Guide	Provides detailed information about how to use Amazon VPC.
Amazon Virtual Private Cloud 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 Virtual Private Cloud User Guide*.

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`.

```
Supported Platforms
VPC
Default VPC
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.

Network ⓘ	vpc-1a2b3c4d (172.31.0.0/16) (default)	 Create new VPC
Subnet ⓘ	No preference (default subnet in any Availability Zor)	Create new subnet

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.

```
Supported Platforms
EC2
VPC
```

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 Virtual Private Cloud User Guide*.

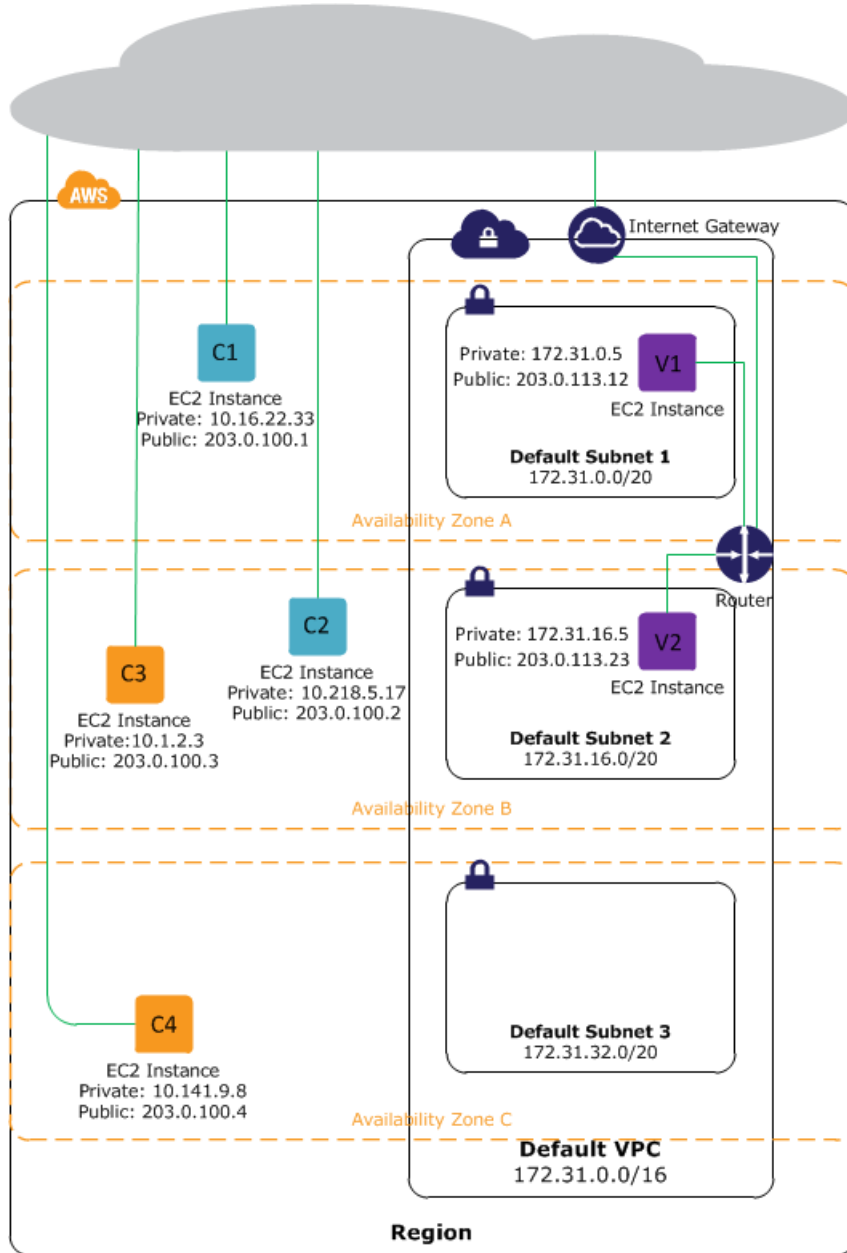
Differences Between Instances in EC2-Classic and EC2-VPC

With EC2-Classic, we assign each instance a private IP address from a shared private IP address range. We also assign each instance a public IP address from Amazon's pool of public IP addresses. Instances access the Internet directly through the AWS network edge.

With EC2-VPC, we assign each instance a private IP address from the private IP address range of your VPC. You can control the IP address range, subnets, routing, network gateways, network ACLs, and security groups for your VPC. You can specify whether your instance receives a public IP address during launch. Instances with public IP addresses or Elastic IP addresses can access the Internet through a logical Internet gateway attached to the AWS network edge. For more information about EC2-VPC, see [What is Amazon VPC?](#) in the *Amazon Virtual Private Cloud User Guide*.

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, and can access other services such as Amazon Simple Storage Service (Amazon S3).
- 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 the following through the Internet gateway: instances in EC2-Classic, other services (such as Amazon S3), and the Internet.



For more information about the differences between EC2-Classic and EC2-VPC, see [Amazon EC2 and Amazon Virtual Private Cloud \(VPC\)](#) (p. 309).

Migrating from EC2-Classic to 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. 312). 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 \(VPC\)](#) (p. 309).

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 in your account can be shared

between EC2-Classic and a VPC; for example, key pairs. However the following resources cannot be shared or moved between platforms:

- Security groups
- Elastic IP addresses
- Instances
- Elastic Load Balancers

If your account supports EC2-Classic, you may have already set up a number of resources for use with EC2-Classic. If you want to migrate to using a VPC, you will have to recreate those resources in your VPC. We're working on features to help you migrate your resources.

Note

If you use Reserved Instances, you can change the network platform for your Reserved Instances from EC2-Classic to EC2-VPC.

Migrating to a VPC

This topic provides some basic steps to illustrate how you can migrate to a VPC, and provides an example of migrating a simple web application.

Topics

- [Step 1: Create a VPC \(p. 315\)](#)
- [Step 2: Configure Your Security Group \(p. 316\)](#)
- [Step 3: Create an AMI from Your EC2-Classic Instance \(p. 316\)](#)
- [Step 4: Launch an Instance Into Your VPC \(p. 317\)](#)
- [Example: Migrating a Simple Web Application \(p. 318\)](#)

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 \(p. 76\)](#). 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 \(p. 264\)](#).

To copy your security group rules to a new security group

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Security Groups**.
3. Select the security group that's associated with your EC2-Classic instance, then click **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 \(p. 264\)](#).

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. Click **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. 62) .
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. 64) .

(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 \(p. 353\)](#)
- [Creating an Amazon EBS Volume \(p. 357\)](#)
- [Attaching an Amazon EBS Volume to an Instance \(p. 361\)](#)

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 \(p. 381\)](#)
- [Creating an Amazon EBS Snapshot \(p. 382\)](#)
- [Restoring an Amazon EBS Volume from a Snapshot \(p. 359\)](#)

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.
2. On the dashboard, click **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 click **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 click 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. Click **Review and Launch**.
7. Review your instance details, then click **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 \(p. 130\)](#).

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. In the account in which you created your AMI, open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. In the **Filter** list, ensure **Owned by me** is selected, then select your AMI.
4. In the **Permissions** tab, click **Edit**. Enter the account number of your new AWS account, click **Add Permission**, and then click **Save**.

To launch an instance into your default VPC

1. In your new AWS account, open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. In the **Filter** list, select **Private images**. Select the AMI that you shared from your EC2-Classic account, then click **Launch**.
4. On the **Choose an Instance Type** page, select the type of instance, and click **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, your default VPC should be selected in the **Network** list. Configure any other details you require, then click 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. Click **Review and Launch**.
7. Review your instance details, then click **Launch** to specify a key pair and launch your instance.

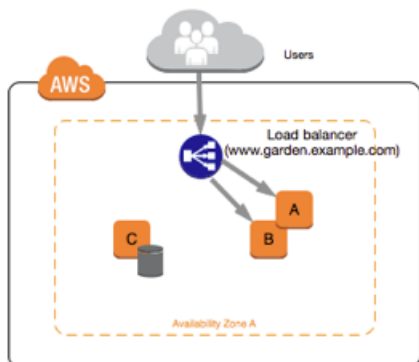
For more information about the parameters you can configure in each step of the wizard, see [Launching an Instance](#) (p. 130).

Example: Migrating a Simple Web Application

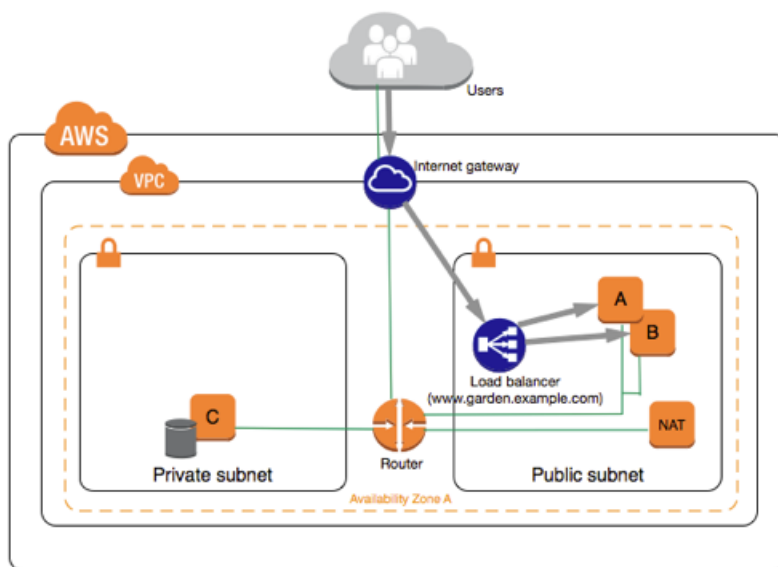
In this example, you use AWS to host your gardening website. To manage your website, you have three running instances in EC2-Classic. Instances A and B host your public-facing web application, and you use an Elastic Load Balancer to load balance the traffic between these instances. You've assigned Elastic 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.

Amazon Elastic Compute Cloud User Guide for Microsoft Windows

Migrating from EC2-Classic to a VPC



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 assign new Elastic IP addresses to your web servers, 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 Virtual Private Cloud 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. 316).
- **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. 316).

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 \(p. 317\)](#).
- **Create new Elastic IP addresses:** You cannot use your EC2-Classic Elastic IP addresses in a VPC. Instead, create new Elastic IP addresses for use in a VPC, and assign them to your web servers. For more information, see [Elastic IP Addresses \(EIP\) \(p. 329\)](#).
- **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.

Note

Currently, there is no process that allows you to connect your EC2-Classic instances directly to instances in your VPC; for example, to synchronize data between them. We are working on tools to help you with this task. In the meantime, you may want to explore options such as SSH tunneling to achieve this.

- **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.

- **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.

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 \(p. 312\)](#). For information about Amazon VPC, see [What is Amazon VPC?](#) in the *Amazon Virtual Private Cloud User Guide*.

Contents

- [Private Addresses and Internal DNS Hostnames \(p. 321\)](#)
- [Public IP Addresses and External DNS Hostnames \(p. 321\)](#)
- [Differences Between EC2-Classic and EC2-VPC \(p. 322\)](#)
- [Determining Your Public, Private, and Elastic IP Addresses \(p. 323\)](#)
- [Assigning a Public IP Address \(p. 324\)](#)
- [Multiple Private IP Addresses \(p. 325\)](#)

Private Addresses and Internal DNS Hostnames

You can use private IP addresses and internal DNS hostnames for communication between instances in the same network (EC2-Classic or a VPC). Private IP addresses are not reachable from the Internet. For more information about private IP addresses, see [RFC 1918](#).

When you launch an instance, we allocate a private IP address for the instance using DHCP.

Each instance that you launch into a VPC has a default network interface. The network interface specifies the primary private IP address for the instance. If you don't select a primary private IP address, we select an available IP address in the subnet's range. You can 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 \(p. 325\)](#).

Each instance is provided an internal DNS hostname that resolves to the private IP address of the instance in EC2-Classic or your VPC. We can't resolve the DNS hostname outside the network that the instance is in.

If you create a custom firewall configuration in EC2-Classic, you must allow inbound traffic from port 53 (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 on Windows, use the following command: **ipconfig /all | findstr /c:"DNS Servers"**.

For instances launched in EC2-Classic, a private IP address is associated with the instance until it is stopped or terminated.

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.

Public IP Addresses and External DNS Hostnames

You can use public IP addresses and external DNS hostnames for communication between your instances and the Internet or other AWS products, such as Amazon Simple Storage Service (Amazon S3). Public IP addresses are reachable from the Internet.

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 EC2-VPC, you can control whether your instance receives a public IP address. The public IP address is assigned to the eth0 network interface (the primary network interface).

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. Therefore, if you want an instance in a nondefault subnet to communicate with the Internet, you must either enable the public IP addressing

feature during launch, or associate an Elastic IP address with the primary or any secondary private IP address assigned to the network interface for the instance. You can also modify the public IP addressing attribute of a nondefault subnet to specify that instances that are launched into that subnet should receive a public IP address. For more information, see [Modifying Your Subnet's Public IP Addressing Behavior](#) in the *Amazon Virtual Private Cloud User Guide*.

Note

T2 instance types can only be launched into a VPC. If you use the Amazon EC2 launch wizard to launch a T2 instance type in your EC2-Classic account, and you have no VPCs, the launch wizard creates a nondefault VPC for you, and modifies the subnet's attribute to automatically request a public IP address for your instance. For more information about T2 instance types, see [T2 Instances \(p. 76\)](#).

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 (EIP) with your instance, or when you associate an EIP with the primary network interface (eth0) of your instance in a VPC. When you disassociate the EIP 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 (EIP) instead. You can allocate your own EIP, and associate it to your instance. For more information, see [Elastic IP Addresses \(EIP\) \(p. 329\)](#).

We provide each instance that has a public IP address with an external DNS hostname. 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. 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 Virtual Private Cloud User Guide*.

The private IP address and public IP address for an instance are directly mapped to each other through network address translation (NAT). For more information about NAT, see [RFC 1631: The IP Network Address Translator \(NAT\)](#).

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.

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.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Determining Your Public, Private, and Elastic IP Ad-
resses**

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 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.
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 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.

Determining Your Public, Private, and Elastic IP Addresses

You can use the EC2 console to determine the private IP addresses, public IP addresses, and EIPs of your instances.

To determine your instance's IP addresses using the console

1. Open the Amazon EC2 console.
2. Click **Instances** in the navigation pane.
3. Select an instance. The console displays information about the instance in the lower pane.
4. Get the public IP address from the **Public IP** field.
5. If an EIP has been associated with the instance, get the EIP from the **Elastic IP** field.
6. Get the private IP address from the **Private IP** field.

You can also determine the public and private IP addresses of your instances using instance metadata. For more information, see [Instance Metadata and User Data \(p. 100\)](#).

To determine your instance's IP addresses using instance metadata

1. Connect to the instance.

2. Use the following command to access the private IP address:

```
C:\> GET http://169.254.169.254/latest/meta-data/local-ipv4
```

3. Use the following command to access the public IP address:

```
C:\> GET http://169.254.169.254/latest/meta-data/public-ipv4
```

Note that if an EIP is associated with the instance, the value returned is that of the EIP.

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 \(p. 321\)](#). 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 \(EIP\) \(p. 329\)](#).

To access the public IP addressing feature when launching an instance

1. Open the Amazon EC2 console.
2. Click **Launch Instance**.
3. Choose an AMI and click its **Select** button, then choose an instance type and click **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. Select **Enable** or **Disable** to override the default setting for the subnet.

The following rules apply:

- A public IP address can only be assigned to a single 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, and is not available for the eth1 network interface.
 - You can only assign a public IP address to a new network interface, not an existing one.
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 \(p. 130\)](#). On the final **Review Instance Launch** page, review your settings, and then click **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 \(EIP\)](#) (p. 329). 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 <code>--associate-public-ip-address</code> or the <code>--no-associate-public-ip-address</code> option with the run-instances command.
Amazon EC2 CLI	Use the <code>--associate-public-ip-address</code> option with the ec2-run-instances command.
AWS Tools for Windows PowerShell	Use the <code>-AssociatePublicIp</code> parameter with the New-EC2Instance command.
Query API	Use the <code>NetworkInterface.n.AssociatePublicIpAddress</code> 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. 335).

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. 325)
- [Assigning a Secondary Private IP Address](#) (p. 326)
- [Configuring the Operating System on Your Instance to Recognize the Secondary Private IP Address](#) (p. 328)
- [Associating an Elastic IP Address with the Secondary Private IP Address](#) (p. 328)
- [Viewing Your Secondary Private IP Addresses](#) (p. 328)
- [Unassigning a Secondary Private IP Address](#) (p. 329)

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.
2. Click the **Launch Instance** button.
3. Choose an AMI and click its **Select** button, then choose an instance type and click **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 click **Next: Add Storage**:
 - a. Click **Add Device** to add another network interface. The console enables you specify up to 2 network interfaces when you launch an instance. After you launch the instance, click **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 \(p. 335\)](#).
 - 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**, click **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. 328).

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 click **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. Click **Review and Launch**.
9. On the **Review Instance Launch** page, review your settings, and then click **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.
2. In the navigation pane, click **Network Interfaces**, and then right-click the network interface attached to the instance.
3. Select **Manage Private IP Addresses**.
4. In the **Manage Private IP Addresses** dialog box, do the following:
 - a. Click **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. Click **Yes, Update**, and then click **Close**.

Note

You can also assign a secondary private IP address to an instance by clicking **Instances** in the navigation pane, right-clicking your instance, and selecting **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.

For information about configuring a Windows instance, see [Configuring a Secondary Private IP Address for Your Windows Instance in a VPC](#) (p. 179).

Associating an Elastic IP Address with the Secondary Private IP Address

To associate an EIP with a secondary private IP address in EC2-VPC

1. Open the Amazon EC2 console.
2. Click **Elastic IPs** in the navigation pane.
3. Right-click the IP address, and then click **Associate**.
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** drop-down list.
5. Click **Associate**.

To associate an EIP 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.
2. Click **Network Interfaces** in the navigation pane.
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.
2. Click **Instances** in the navigation pane.
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.
2. Click **Instances** in the navigation pane.
3. Right-click an instance, and then click **Manage Private IP Addresses**.
4. In the **Manage Private IP Addresses** dialog box, beside the secondary private IP address to unassign, click **Unassign**.
5. Click **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.
2. Click **Network Interface** in the navigation pane.
3. Right-click an instance, and then click **Manage Private IP Addresses**.
4. In the **Manage Private IP Addresses** dialog box, beside the secondary private IP address to unassign, click **Unassign**.
5. Click **Yes, Update**, and then click **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 (EIP)

An *Elastic IP address* (EIP) is a static IP address designed for dynamic cloud computing. With an EIP, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account. Your EIP is associated with your AWS account, not a particular instance, and it remains associated with your account until you choose to explicitly release it.

There's one pool of EIPs for use with the EC2-Classic platform and another for use with your VPC. You can't associate an EIP that you allocated for use with a VPC with an instance in EC2-Classic, and vice-versa. For more information about EC2-Classic and EC2-VPC, see [Supported Platforms \(p. 312\)](#).

Topics

- [Elastic IP Addresses in EC2-Classic \(p. 330\)](#)
- [Elastic IP Addresses in a VPC \(p. 330\)](#)

- [Differences Between EC2-Classic and EC2-VPC](#) (p. 331)
- [Allocating an Elastic IP Address](#) (p. 331)
- [Describing Your Elastic IP Addresses](#) (p. 332)
- [Associating an Elastic IP Address with a Running Instance](#) (p. 332)
- [Associating an Elastic IP Address with a Different Running Instance](#) (p. 333)
- [Releasing an Elastic IP Address](#) (p. 333)
- [Using Reverse DNS for Email Applications](#) (p. 334)
- [Elastic IP Address Limit](#) (p. 334)

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 EIP.

When you associate an EIP with an instance, the instance's current public IP address is released to the EC2-Classic public IP address pool. If you disassociate an EIP 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 EIP from it.

To ensure efficient use of EIPs, we impose a small hourly charge if an EIP 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 EIP with the elastic network interface (ENI) attached to the instance.

When you associate an EIP 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 EIP 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 EIP with it manually. The EIP remains associated with the instance when you stop it.

To ensure efficient use of EIPs, we impose a small hourly charge if an EIP 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 EIP associated with the instance, but you are charged for any additional EIPs associated with the instance. For more information, see [Amazon EC2 Pricing](#).

For information about using an EIP with an instance in a VPC, see [Elastic IP Addresses](#) in the *Amazon Virtual Private Cloud User Guide*.

Differences Between EC2-Classical and EC2-VPC

The following table lists the differences between EIPs on EC2-Classical and EC2-VPC.

Characteristic	EC2-Classical	EC2-VPC
Allocation	When you allocate an EIP, it's for use only in EC2-Classical.	When you allocate an EIP, it's for use only in a VPC.
Association	You associate an EIP with an instance.	An EIP is a property of an elastic network interface (ENI). You can associate an EIP with an instance by updating the ENI attached to the instance. For more information, see Elastic Network Interfaces (ENI) (p. 334).
Reassociation	If you try to associate an EIP 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 EIP 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-Classical account, and you try to associate an EIP that's already associated with another instance, it succeeds only if you allowed reassociation.
Instance stop	If you stop an instance, its EIP is disassociated, and you must re-associate the EIP when you restart the instance.	If you stop an instance, its EIP remains associated.
Multiple IP	Instances support only a single private IP address and a corresponding EIP.	Instances support multiple IP addresses, and each one can have a corresponding EIP. For more information, see Multiple Private IP Addresses (p. 325).

Allocating an Elastic IP Address

You can allocate an Elastic IP address using the AWS Management Console or the command line.

To allocate an Elastic IP address for use with EC2-Classical using the console

1. Open the Amazon EC2 console.
2. Click **Elastic IPs** in the navigation pane.
3. Click **Allocate New Address**.
4. Select `EC2` from the **EIP** list, and then click **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 AWS Management Console or the command line.

To describe your Elastic IP addresses using the console

1. Open the Amazon EC2 console.
2. Click **Elastic IPs** in the navigation pane.
3. To filter the displayed list, start typing part of the EIP or the ID of the instance to which it is assigned in the search box.

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 AWS Management Console or the command line.

To associate an Elastic IP address with an instance using the console

1. Open the Amazon EC2 console.
2. Click **Elastic IPs** in the navigation pane.
3. Select an EIP and click **Associate Address**.
4. In the **Associate Address** dialog box, select the instance from the **Instance** list box and click **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)

Associating an Elastic IP Address with a Different Running Instance

You can reassociate an Elastic IP address using the AWS Management Console or the command line.

To reassociate an Elastic IP address using the console

1. Open the Amazon EC2 console.
2. Click **Elastic IPs** in the navigation pane.
3. Select the EIP, and then click the **Disassociate** button.
4. Click **Yes, Disassociate** when prompted.
5. Select the EIP, and then click **Associate**.
6. In the **Associate Address** dialog box, select the new instance from the **Instance** list, and then click **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)

Releasing an Elastic IP Address

If you no longer need an EIP, we recommend that you release it (the address must not be associated with an instance). You incur charges for any EIP that's allocated for use with EC2-Classical but not associated with an instance.

You can release an Elastic IP address using the AWS Management Console or the command line.

To release an Elastic IP address using the console

1. Open the Amazon EC2 console.
2. Click **Elastic IPs** in the navigation pane.
3. Select the Elastic IP address, click the **Release Address** button, and then click **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 in the [Request to Remove Email Sending Limitations form](#). AWS works with ISPs and Internet anti-spam organizations (such as Spamhaus) 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 (A Record) pointing to your Elastic IP address must exist before we can create your reverse DNS record.

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 EIP primarily for load balancing use cases, and use DNS hostnames for all other inter-node communication.

If you feel your architecture warrants additional EIPs, 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 network interface for eth0 when you launch an instance, but only when you create a 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 a network interface, attach it to an instance, detach it from an instance, and attach it to another instance. The attributes of a network interface follow the network interface as it is attached or

detached from an instance and reattached to another instance. When you move a network interface from one instance to another, network traffic is redirected to the new instance.

Each instance in a VPC has a default network interface. The default network interface has a primary private IP address in the IP address range of its VPC. You can create and attach additional network interfaces. The maximum number of network interfaces that you can use varies by instance type. For more information, see [Private IP Addresses Per ENI Per Instance Type \(p. 335\)](#).

Attaching multiple 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 \(p. 335\)](#)
- [Creating a Management Network \(p. 337\)](#)
- [Use Network and Security Appliances in Your VPC \(p. 337\)](#)
- [Creating Dual-homed Instances with Workloads/Roles on Distinct Subnets \(p. 338\)](#)
- [Create a Low Budget High Availability Solution \(p. 338\)](#)
- [Best Practices for Configuring Network Interfaces \(p. 338\)](#)
- [Creating a Network Interface \(p. 338\)](#)
- [Deleting a Network Interface \(p. 339\)](#)
- [Viewing Details about a Network Interface \(p. 339\)](#)
- [Attaching a Network Interface When Launching an Instance \(p. 340\)](#)
- [Attaching a Network Interface to a Stopped or Running Instance \(p. 341\)](#)
- [Detaching a Network Interface from an Instance \(p. 342\)](#)
- [Changing the Security Group of a Network Interface \(p. 342\)](#)
- [Changing the Source/Destination Checking of a Network Interface \(p. 343\)](#)
- [Associating an Elastic IP Address with a Network Interface \(p. 343\)](#)
- [Disassociating an Elastic IP Address from a Network Interface \(p. 344\)](#)
- [Changing Termination Behavior for a Network Interface \(p. 344\)](#)
- [Adding or Editing a Description for a Network Interface \(p. 345\)](#)
- [Adding or Editing Tags for a Network Interface \(p. 345\)](#)

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 \(p. 325\)](#).

Instance Type	Maximum Interfaces	IP Addresses per Interface
c1.medium	2	6
c1.xlarge	4	15
c3.large	3	10

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Private IP Addresses Per ENI Per Instance Type**

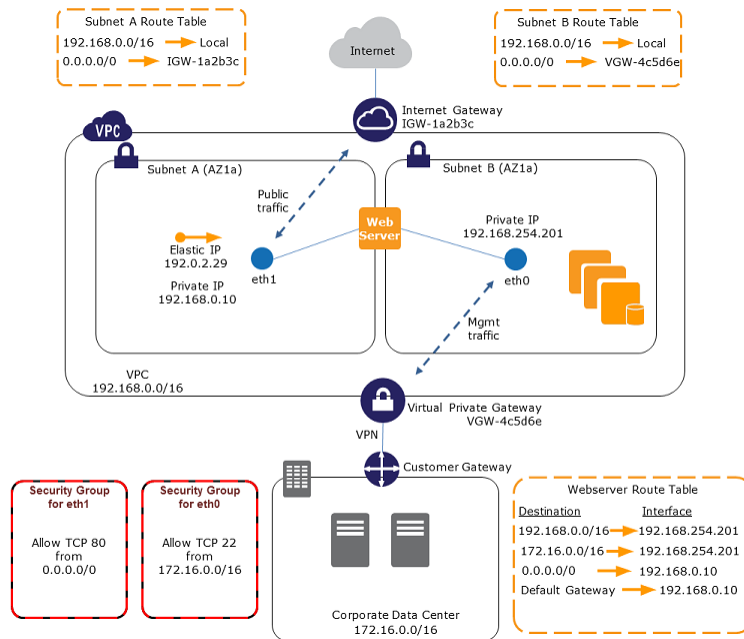
Instance Type	Maximum Interfaces	IP Addresses per Interface
c3.xlarge	4	15
c3.2xlarge	4	15
c3.4xlarge	8	30
c3.8xlarge	8	30
cc2.8xlarge	8	30
cg1.4xlarge	8	30
cr1.8xlarge	8	30
g2.2xlarge	4	15
hi1.4xlarge	8	30
hs1.8xlarge	8	30
i2.xlarge	4	15
i2.2xlarge	4	15
i2.4xlarge	8	30
i2.8xlarge	8	30
m1.small	2	4
m1.medium	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
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

Instance Type	Maximum Interfaces	IP Addresses per Interface
t2.small	2	4
t2.medium	3	6

Creating a Management Network

You can create a management network using network interfaces. In this scenario, the secondary network interface on the instance handles public-facing traffic and the primary 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 a 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 network interfaces. You can create and attach secondary 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 a 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 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.

Best Practices for Configuring Network Interfaces

- You can attach a 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*) network interfaces when the instance is running or stopped. However, you can't detach the primary (eth0) interface.
- You can attach a network interface in one subnet to an instance in another subnet in the same VPC, however, both the 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 network interfaces to attach to the instance for both the primary (eth0) and additional network interfaces.
- Launching an 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 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 Microsoft Windows Server automatically recognize the warm or hot attach and configure themselves.)
- Attaching another network interface to an instance is not a method to increase or double the network bandwidth to or from the dual-homed instance.

Creating a Network Interface

You can create a network interface using the AWS Management Console or the command line.

To create a network interface using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Click **Create Network Interface**.
4. In the **Create Network Interface** dialog box, provide the following information for the network interface, and then click **Yes, Create**.

- a. In **Description**, enter a descriptive name.
- b. In **Subnet**, select the subnet. Note that you can't move the network interface to another subnet after it's created, and you can only attach the network 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 a 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 a Network Interface

You must first detach a network interface from an instance before you can delete it. Deleting a network interface releases all attributes associated with the network interface and releases any private IP addresses or Elastic IP addresses to be used by another instance.

You can delete a network interface using the AWS Management Console or the command line.

To delete a network interface using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Select a network interface, and then click the **Delete** button.
4. In the **Delete Network Interface** dialog box, click **Yes, Delete**.

To delete a 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 a Network Interface

You can describe a network interface using the AWS Management Console or the command line.

To describe a network interface using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Select the network interface.

4. View the details on the **Details** tab.

To describe a 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 a 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 a Network Interface When Launching an Instance

You can attach an additional network interface to an instance when you launch it into a VPC.

Note

If an error occurs when attaching a network interface to your instance, this causes the instance launch to fail.

You can attach a network interface to an instance using the AWS Management Console or the command line.

To attach a network interface when launching an instance using the console

1. Open the Amazon EC2 console.
2. Click **Launch Instance**.
3. Choose an AMI and click its **Select** button, then choose an instance type and click **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 network interface for the primary network interface (eth0) or multiple network interfaces in the next step.

5. In the **Network Interfaces** section, the console enables you specify up to 2 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 network interface. When you've finished, click **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 \(p. 335\)](#).

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Attaching a Network Interface to a Stopped or Running
Instance**

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 click **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. Click **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 click **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 a 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 a Network Interface to a Stopped or Running Instance

You can attach a 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.

To attach a network interface to an instance using the Instances page

1. Open the Amazon EC2 console.
2. Click **Instances** in the navigation pane.
3. Right-click the instance, and then select **Attach Network Interface**.
4. In the **Attach Network Interface** dialog box, select the network interface, and then click **Attach**.

To attach a network interface to an instance using the Network Interfaces page

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Select the network interface.
4. Click the **Attach** button.
5. In the **Attach Network Interface** dialog box, select the instance, and then click **Attach**.

To attach a 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 a Network Interface from an Instance

You can detach a secondary 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 a network interface from an instance using the Instances page

1. Open the Amazon EC2 console.
2. Click **Instances** in the navigation pane.
3. Right-click the instance, and then select **Detach Network Interface**.
4. In the **Detach Network Interface** dialog box, select the network interface, and then click **Detach**.

To detach a network interface from an instance using the Network Interfaces page

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Select the network interface, and then click the **Detach** button.
4. In the **Detach Network Interface** dialog box, click **Yes, Detach**. If the network interface fails to detach from the instance, select **Force detachment**, and then try again.

To detach a 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 a Network Interface

You can change the security groups that are associated with a network interface. When you create the security group, be sure to specify the same VPC as the subnet for the network interface.

You can change the security group for your network interfaces using the AWS Management Console 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 a network interface using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Select the network interface.
4. Right-click the network interface, and then select **Change Security Groups**.
5. In the **Change Security Groups** dialog box, select the security groups to use, and then click **Save**.

To change the security group of a 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 a 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 AWS Management Console or the command line.

To change source/destination checking for a network interface using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Right-click the network interface, and then select **Change Source/Dest Check**.
4. In the dialog box, select **Enabled** (if enabling), or **Disabled** (if disabling), and then click **Save**.

To change source/destination checking for a 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 a Network Interface

If you have an Elastic IP address, you can associate it with one of the private IP addresses for the network interface. You can associate one Elastic IP address with each private IP address.

You can associate an Elastic IP address using the AWS Management Console or the command line.

To associate an Elastic IP address using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Right-click the network interface, 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 click **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 a Network Interface

If the network interface has an Elastic IP address associated with it, you can disassociate the address, and then either associate it with another 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 a network interface, as network interfaces are specific to a particular subnet.

You can disassociate an Elastic IP address using the AWS Management Console or the command line.

To disassociate an Elastic IP address using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Right-click the network interface, and then select **Disassociate Address**.
4. In the **Disassociate IP Address** dialog box, click **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 a Network Interface

You can set the termination behavior for a network interface attached to an instance so that it is automatically deleted when you delete the instance it's attached to.

Note

By default, network interfaces that are automatically created and attached to instances using the AWS Management 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 a network interface using the AWS Management Console or the command line.

To change the termination behavior for a network interface using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Right-click the network interface, 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 network interface to be deleted when you terminate an instance.

To change the termination behavior for a 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 a Network Interface

You can change the description for a network interface using the AWS Management Console or the command line.

To change the description for a network interface using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Right-click the network interface, and then select **Change Description**.
4. In the **Change Description** dialog box, enter a description for the network interface, and then click **Save**.

To change the description for a 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 a Network Interface

Tags are metadata that you can add to a 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 \(p. 427\)](#).

You can tag a resource using the AWS Management Console or the command line.

To add or edit tags for a network interface using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Select the network interface.
4. In the details pane, click the **Tags** tab, and then click **Add/Edit Tags**.
5. In the **Add/Edit Tags** dialog, click **Create Tag** for each tag you want to create, and enter a key and optional value. When you're done, click **Save**.

To add or edit tags for a 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)

Enabling Enhanced Networking on Windows Instances in a VPC

With C3, R3, and I2 instances, you can enable enhanced networking capabilities. Amazon EC2 supports enhanced networking capabilities using single root I/O virtualization (SR-IOV). 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

- [Requirements \(p. 346\)](#)
- [Testing Whether Enhanced Networking Is Enabled \(p. 347\)](#)
- [Enabling Enhanced Networking on Windows \(p. 348\)](#)

Note that you can get directions for Linux from [Enabling Enhanced Networking on Linux Instances in a VPC](#) in the *Amazon Elastic Compute Cloud User Guide for Linux*.

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 one of the following instance types: `c3.large`, `c3.xlarge`, `c3.2xlarge`, `c3.4xlarge`, `c3.8xlarge`, `i2.xlarge`, `i2.2xlarge`, `i2.4xlarge`, `i2.8xlarge`, `r3.large`,

`r3.xlarge`, `r3.2xlarge`, `r3.4xlarge`, or `r3.8xlarge`. For more information about instance types, see [Amazon EC2 Instances](#).

- 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 sriovNetSupport
```

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\vxm62x64.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).
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 \(p. 62\)](#). The AMI inherits the enhanced networking attribute from the instance. Therefore, you can use this AMI to launch another C3, R3, or I2 instance with the 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).

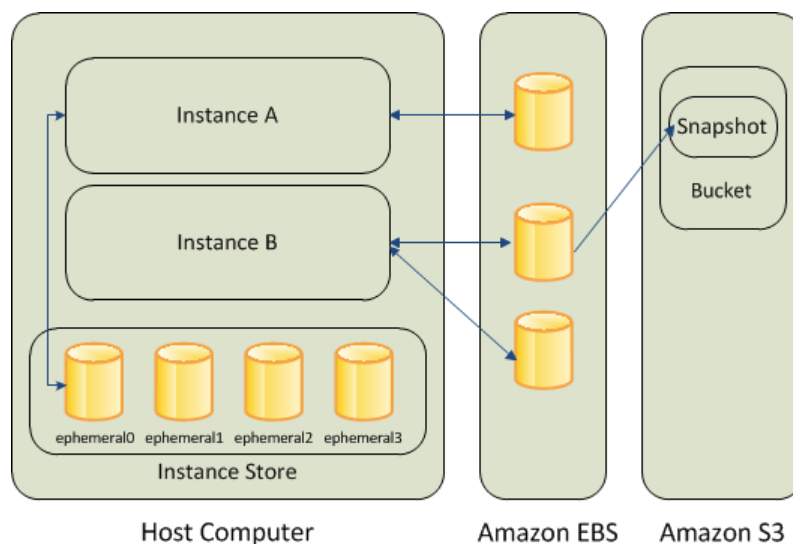
Storage

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 Elastic Compute Cloud to meet your specific requirements. These storage options include the following:

- [Amazon Elastic Block Store \(Amazon EBS\)](#) (p. 351)
- [Amazon EC2 Instance Store](#) (p. 403)
- [Amazon Simple Storage Service \(Amazon S3\)](#) (p. 409)

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 Amazon EC2 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 Amazon 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 Amazon EC2 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. Amazon EBS volumes can also be created as encrypted volumes using the Amazon EBS encryption feature. For more information, see [Amazon EBS Encryption \(p. 387\)](#).

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. For more information, see [Amazon Elastic Block Store \(Amazon EBS\) \(p. 351\)](#).

Amazon EC2 Instance Store

Many Amazon EC2 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 Amazon EC2 instances. The data on an instance store volume persists only during the life of the associated Amazon EC2 instance; if you stop or terminate an instance, any data on instance store volumes is lost. For more information, see [Amazon EC2 Instance Store \(p. 403\)](#).

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 \(Amazon S3\) \(p. 409\)](#).

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 \(p. 411\)](#).

You can also attach EBS volumes to a running instance. For more information, see [Attaching an Amazon EBS Volume to an Instance \(p. 361\)](#).

Amazon Elastic Block Store (Amazon EBS)

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with Amazon EC2 instances. Amazon EBS volumes are highly available and reliable storage volumes that can be attached to any running instance that is in the same Availability Zone. Amazon EBS volumes that are attached to an Amazon 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. Amazon 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 Amazon 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. 387).

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 Amazon 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. 352)
- [Amazon EBS Volumes](#) (p. 353)
- [Amazon EBS Snapshots](#) (p. 381)
- [Amazon EBS Encryption](#) (p. 387)
- [Amazon EBS Volume Performance](#) (p. 389)
- [Amazon EBS API and Command Overview](#) (p. 401)

Features of Amazon EBS

- You can create Amazon EBS storage volumes from 1 GiB to 1 TiB in size and mount them 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. 357).
- 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. For more information, see [General Purpose \(SSD\) Volumes](#) (p. 355).
- With Provisioned IOPS (SSD) volumes, you can provision a specific level of I/O performance, up to 4000 IOPS per volume. This allows you to predictably scale to thousands of IOPS per EC2 instance. For more information, see [Provisioned IOPS \(SSD\) Volumes](#) (p. 357).
- Amazon 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. 363).
- You can use encrypted Amazon 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. 387).
- You can create point-in-time snapshots of Amazon 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 Amazon 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. 381).
- Amazon 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. 382), [Restoring an Amazon EBS Volume from a Snapshot](#) (p. 359), and [Copying an Amazon EBS Snapshot](#) (p. 384).

- A large repository of public data set snapshots can be restored to Amazon EBS volumes and seamlessly integrated into AWS cloud-based applications. For more information, see [Using Public Data Sets \(p. 421\)](#).
- 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 \(p. 389\)](#).

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 Amazon 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. Amazon 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 \(p. 355\)](#).

Contents

- [Benefits of Using Amazon EBS Volumes \(p. 353\)](#)
- [Amazon EBS Volume Types \(p. 355\)](#)
- [Creating an Amazon EBS Volume \(p. 357\)](#)
- [Restoring an Amazon EBS Volume from a Snapshot \(p. 359\)](#)
- [Attaching an Amazon EBS Volume to an Instance \(p. 361\)](#)
- [Making an Amazon EBS Volume Available for Use \(p. 363\)](#)
- [Viewing Volume Information \(p. 365\)](#)
- [Monitoring the Status of Your Volumes \(p. 365\)](#)
- [Detaching an Amazon EBS Volume from an Instance \(p. 375\)](#)
- [Deleting an Amazon EBS Volume \(p. 376\)](#)
- [Expanding the Storage Space of a Volume \(p. 377\)](#)

Benefits of Using Amazon EBS Volumes

Data Availability

When you create an Amazon 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 Amazon 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 Amazon EBS volume with a file system, such as NTFS, and then install applications.

An Amazon 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 Amazon EBS volumes at no additional charge (this includes data for the root device volumes for Amazon EBS-backed instances). For more information, see [Monitoring Volumes with CloudWatch \(p. 365\)](#).

Data Persistence

An Amazon 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, Amazon 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 Amazon 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 Amazon 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 \(p. 387\)](#).

By default, Amazon 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 Amazon EBS volumes with the Amazon EBS encryption feature. You can use encrypted Amazon 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 Amazon EC2 instance, providing encryption of data-in-transit from the EC2 instance to EBS storage. For more information, see [Amazon EBS Encryption \(p. 387\)](#).

Snapshots

Amazon EBS provides the ability to create snapshots (backups) of any Amazon EC2 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 to 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 Amazon EBS volumes, expand the size of a volume, or move volumes across Availability Zones. Snapshots of encrypted Amazon 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. Amazon 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.

Amazon EBS 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 \(p. 427\)](#).

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 \(p. 355\)](#)
- [Provisioned IOPS \(SSD\) Volumes \(p. 357\)](#)
- [Magnetic Volumes \(p. 357\)](#)

The following table describes basic use cases and performance characteristics for each volume type.

Characteristic	General Purpose (SSD)	Provisioned IOPS (SSD)	Magnetic
Use cases	<ul style="list-style-type: none"> • System boot volumes • Virtual desktops • Small to medium sized databases • Development and test environments 	<ul style="list-style-type: none"> • Critical business applications that require sustained IOPS performance above 3,000 IOPS • Large database workloads, such as: <ul style="list-style-type: none"> • MongoDB • Microsoft SQL Server • MySQL • PostgreSQL • Oracle 	<ul style="list-style-type: none"> • Cold workloads where data is infrequently accessed • Scenarios where the lowest storage cost is important
Volume size	1 GiB - 1 TiB	10 GiB - 1 TiB	1 GiB - 1 TiB
IOPS performance	Has the ability to burst to 3,000 IOPS maximum, with a base performance of 3 IOPS/GiB	Consistently performs at provisioned level, with 4,000 IOPS maximum	Averages 100 IOPS, with the ability to burst to hundreds of IOPS
API and CLI volume name	gp2	io1	standard

There are several factors that can affect the performance of Amazon EBS volumes, such as instance configuration, I/O characteristics, and workload demand. For more information about getting the most out of your Amazon EBS volumes, see [Amazon EBS Volume Performance \(p. 389\)](#).

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 base performance of 3 IOPS/GiB. General Purpose (SSD) volumes can range in size from 1 GiB to 1 TiB.

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 SSD volumes such as Provisioned IOPS (SSD) and General Purpose (SSD). If you are unable to create an SSD volume (or launch an instance with an 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 General Purpose (SSD) and Provisioned IOPS (SSD) volumes by creating a 1 GiB General Purpose (SSD) volume in that zone.

IO Credits and Burst Performance

General Purpose (SSD) volume performance is governed by volume size, which dictates the base performance level of the volume and how quickly it accumulates I/O credits; larger volumes have higher base 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 base performance is needed. The more credits your volume has for I/O, the more time it can burst beyond its base 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 base performance rate of 3 IOPS per GiB of volume size. For example, a 100 GiB General Purpose (SSD) volume has a base performance of 300 IOPS.

When your volume requires more than the base 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 base performance that is equal or greater than the maximum burst performance, so their I/O credit balance never depletes and they can burst indefinitely. 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).

If your volume uses all of its I/O credit balance, the maximum performance of the volume will remain at the base performance level (the rate at which your volume earns credits) until I/O demand drops below the base level and unused credits are added to the I/O credit balance. The larger a volume is, the greater the base performance is and the faster it replenishes the credit balance.

The table below lists several volume sizes and the associated base 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)	Base 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
250	750	2,400	7,200
500	1,500	3,600	3,600
750	2,250	7,200	2,400
1,000	3,000	Infinite	N/A

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:

$$\text{Burst duration} = \frac{(\text{Credit balance})}{(\text{Burst IOPS}) - 3(\text{Volume size in GiB})}$$

If you notice that your volume performance is frequently limited to the base level (due to an empty I/O credit balance), you should consider using a larger General Purpose (SSD) volume (with a higher base performance level) or switching to a Provisioned IOPS (SSD) volume for workloads that require sustained IOPS performance greater than 3,000 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.

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 SSD volumes such as Provisioned IOPS (SSD) and General Purpose (SSD). If you are unable to create an SSD volume (or launch an instance with an 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 General Purpose (SSD) and Provisioned IOPS (SSD) volumes by creating a 1 GiB General Purpose (SSD) volume in that zone.

A Provisioned IOPS (SSD) volume can range in size from 10 GiB to 1 TiB and you can provision up to 4,000 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.

Magnetic Volumes

Magnetic volumes provide the lowest cost per gigabyte of all Amazon 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 a new Amazon EBS volume that you can then attach to any Amazon EC2 instance within the same Availability Zone. You can choose to create an encrypted Amazon EBS volume, but encrypted volumes can only be attached to selected instance types. For more information, see [Supported Instance Types \(p. 387\)](#).

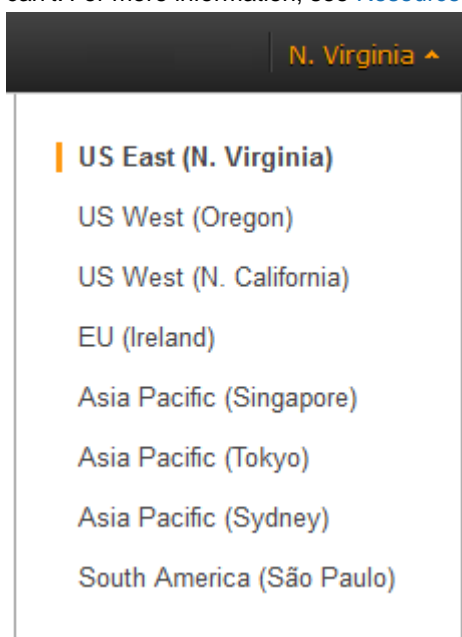
You can also create and attach Amazon EBS volumes when you launch instances by specifying a block device mapping. For more information, see [Launching an Instance \(p. 130\)](#) and [Block Device Mapping \(p. 411\)](#). You can restore volumes from previously created snapshots. For more information, see [Restoring an Amazon EBS Volume from a Snapshot \(p. 359\)](#).

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 \(p. 390\)](#).

When a block of data on a newly created Amazon 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 \(p. 392\)](#).

To create a new Amazon 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 \(p. 424\)](#).



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 \(p. 355\)](#).

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 SSD volumes such as Provisioned IOPS (SSD) and General Purpose (SSD). If you are unable to create an SSD volume (or launch an instance with an 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 General Purpose (SSD) and Provisioned IOPS (SSD) volumes by creating a 1 GiB General Purpose (SSD) volume in that zone.

6. In the **Size** box, enter the size of the volume, in GiB.
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.

Note

Encrypted volumes can only be attached to selected instance types. For more information, see [Supported Instance Types \(p. 387\)](#).

10. Click **Yes, Create**.

To create a new Amazon 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 \(p. 381\)](#).

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 Amazon 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.

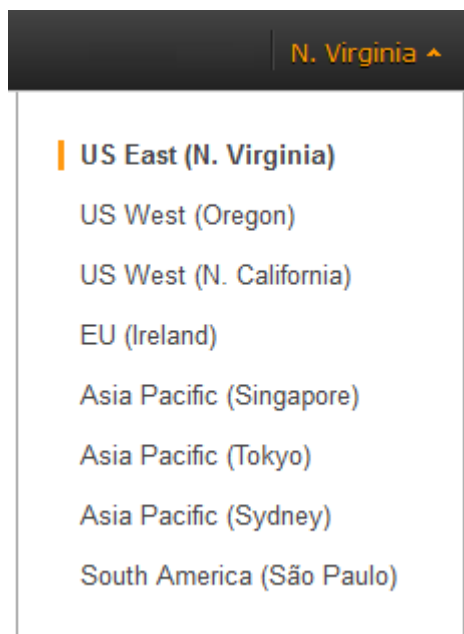
Amazon EBS volumes that are restored from encrypted snapshots are automatically encrypted. Encrypted volumes can only be attached to selected instance types. For more information, see [Supported Instance Types \(p. 387\)](#).

When a block of data on a newly restored Amazon 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 \(p. 392\)](#).

To restore an Amazon EBS volume from a snapshot using the console

You can restore your Amazon 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 \(p. 424\)](#). 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 \(p. 384\)](#).



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 \(p. 355\)](#).

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 SSD volumes such as Provisioned IOPS (SSD) and General Purpose (SSD). If you are unable to create an SSD volume (or launch an instance with an 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 General Purpose (SSD) and Provisioned IOPS (SSD) volumes by creating a 1 GiB General Purpose (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 \(p. 387\)](#).

7. In the **Size** box, enter the size of the volume in GiB, or verify 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. Any AWS Marketplace product codes from the snapshot are propagated to the 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.
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 a Volume \(p. 377\)](#).

To restore a new Amazon 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 your Amazon EBS volume to one of your instances in the same Availability Zone.

To attach an Amazon 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 in the same Availability Zone as the volume are displayed). Encrypted volumes can only be attached to instances that support Amazon EBS encryption. For more information, see [Supported Instance Types \(p. 387\)](#).
5. Verify that the suggested device name is suitable, or enter a device name in the **Device** box. For more information about naming conventions and restrictions, see [Device Naming \(p. 362\)](#).
6. Click **Attach** to attach the volume to the instance. The volume and instance must be in the same Availability Zone.

If a volume has an AWS Marketplace product code:

- The volume can only be attached to the root device of 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 Amazon 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)

Device Naming

The following table lists the available device names on Amazon EC2. You can specify these names when attaching a volume to a running instance or when launching an instance using a block device mapping. The block device driver for the instance assigns the actual volume names when mounting the volumes, and these names can be different than the names that Amazon EC2 recommends. For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 403\)](#). For information about the root device storage, see [Root Device Volume \(p. 8\)](#).

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]

Warning

Although you can attach your Amazon 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 Amazon 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 Amazon EBS volumes per your specifications. For more information, see [Configuring a Windows Instance Using the EC2Config Service \(p. 151\)](#).

Volume Limits

Although Amazon EC2 does not impose limitations on how many volumes you can attach to an instance, there are several factors you need to consider when attaching multiple volumes. The upper limit of volumes you should attach to your instance is dependent on your operating system (Linux instances can reliably handle more volumes than Windows instances). Another consideration is whether you need increased I/O bandwidth or increased storage capacity.

Windows-specific Volume Limits

Windows instances use Red Hat PV, Citrix PV, or AWS PV drivers for storage I/O, and each driver type has its own upper limit for volume attachment. If your Windows instance is using AWS PV or Citrix PV drivers, you can attach up to a total of 25 Amazon EBS volumes using the Amazon EC2 CLI; Windows instances with Red Hat PV drivers are limited to 16 volumes. 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 \(p. 164\)](#).

Although it is technically possible to attach more than 25 volumes to a Windows instance with AWS PV or Citrix PV drivers, this is likely to cause performance issues and is not recommended.

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 the [Amazon EC2 Instance Configuration \(p. 390\)](#) topic 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.

For increased capacity use cases, you can attach many more volumes, but attaching more than 40 volumes (this quantity is only possible on Linux instances; see [Windows-specific Volume Limits \(p. 362\)](#) for Windows instances) might result in boot failures. If you experience boot problems on an instance with many volumes attached, you can stop the instance, detach the non-essential boot volumes, and then reattach them after the instance is running.

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 Amazon 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 Amazon EBS volume and is transparent to applications using the device.

Note that you can take snapshots of your Amazon EBS volume for backup purposes or to use as a baseline when you create another volume. For more information, see [Amazon EBS Snapshots \(p. 381\)](#).

Make the Volume Available on Windows

To use an Amazon EBS volume

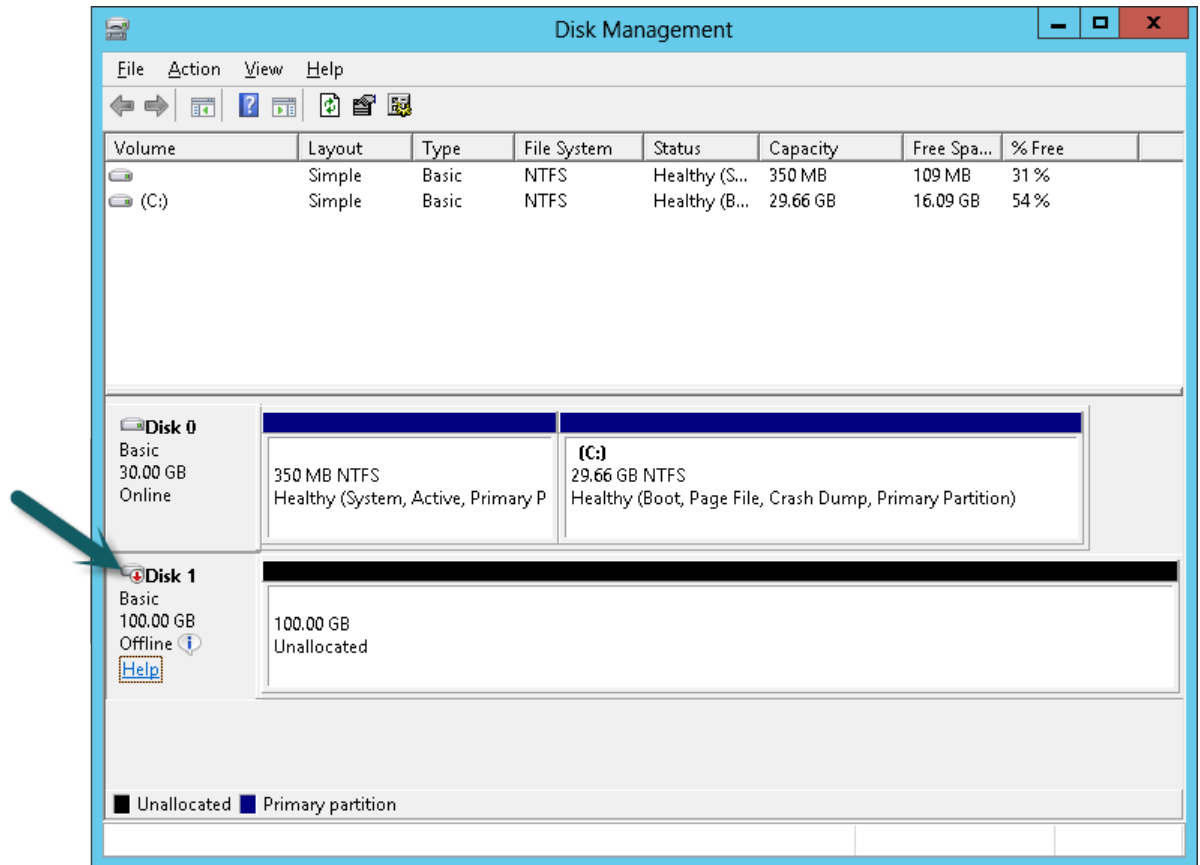
1. Log in to your instance using Remote Desktop.
2. [Windows Server 2012] Go to the Start screen.

[Windows Server 2008] On the taskbar, click **Start**, and then click **Run**.
3. Type **diskmgmt.msc** and press **Enter**. The **Disk Management** utility opens.

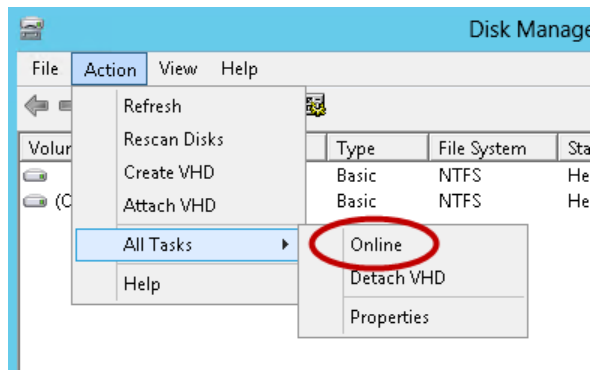
Caution

If you're mounting a volume that already has data on it (for example, a public data set), make sure you don't reformat the volume and delete the existing data.

4. Select the disk that represents the new Amazon EBS volume.



5. On the **Disk Management** menu, select **Action - All Tasks - Online**.



6. A new disk needs to be initialized before it can be used. To initialize the disk:
 - a. In the Disk Management utility, select the new Amazon 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 all of your 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, and the specific instance to which the volume is attached.

To view information about an Amazon 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 Amazon 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 \(p. 365\)](#)
- [Monitoring Volumes with Status Checks \(p. 368\)](#)
- [Monitoring Volume Events \(p. 370\)](#)
- [Working with an Impaired Volume \(p. 371\)](#)
- [Working with the AutoEnableIO Volume Attribute \(p. 373\)](#)

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.
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	The total number of bytes transferred in a specified period of time. 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
VolumeQueueLength	The number of read and write operation requests waiting to be completed in a specified period of time. Units: Count

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
EBS Volumes**

Metric	Description
VolumeThroughputPercentage	<p>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.</p> <p>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).</p> <p>Units: Percent</p>
VolumeConsumedReadWriteOps	<p>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.</p> <p>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.</p> <p>Units: Count</p>

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)	$\text{Sum}(\text{VolumeReadBytes}) / \text{Period} / 1024$
Write Bandwidth (KiB/s)	$\text{Sum}(\text{VolumeWriteBytes}) / \text{Period} / 1024$
Read Throughput (Ops/s)	$\text{Sum}(\text{VolumeReadOps}) / \text{Period}$
Write Throughput (Ops/s)	$\text{Sum}(\text{VolumeWriteOps}) / \text{Period}$
Avg Queue Length (ops)	$\text{Avg}(\text{VolumeQueueLength})$
% Time Spent Idle	$\text{Sum}(\text{VolumeldleTime}) / \text{Period} * 100$
Avg Read Size (KiB/op)	$\text{Avg}(\text{VolumeReadBytes}) / 1024$
Avg Write Size (KiB/op)	$\text{Avg}(\text{VolumeWriteBytes}) / 1024$
Avg Read Latency (ms/op)	$\text{Avg}(\text{VolumeTotalReadTime}) * 1000$
Avg Write Latency (ms/op)	$\text{Avg}(\text{VolumeTotalWriteTime}) * 1000$

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 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 `chkdsk`, 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 following table lists statuses for Amazon EBS volumes.

Overall Volume Status	I/O Enabled Status	I/O Performance Status (Provisioned IOPS (SSD) volumes only)
<code>ok</code>	Enabled (I/O Enabled or I/O Auto-Enabled)	Normal (Volume performance is as expected)
<code>warning</code>	Enabled (I/O Enabled or I/O Auto-Enabled)	Degraded (Volume performance is below expectations) Severely Degraded (Volume performance is well below expectations)

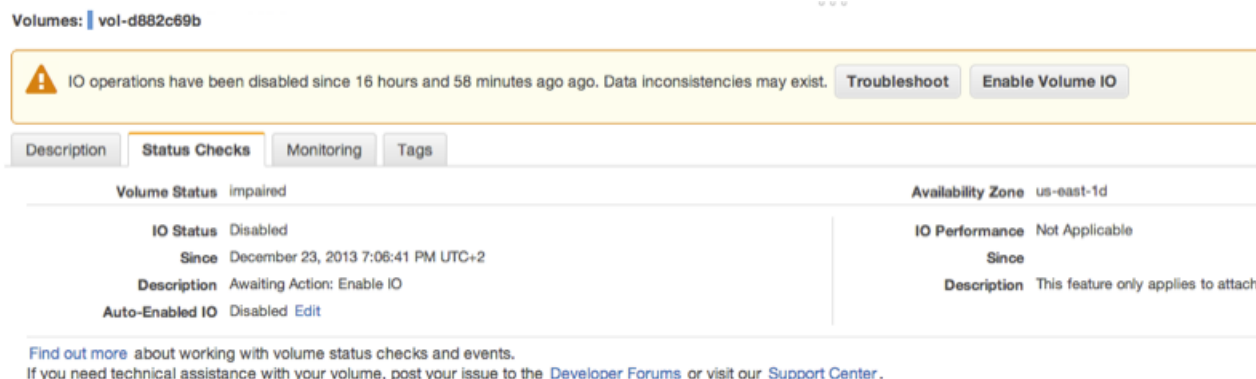
**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
EBS Volumes**

Overall Volume Status	I/O Enabled Status	I/O Performance Status (Provisioned IOPS (SSD) volumes only)
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 \(p. 371\)](#).

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 \(p. 370\)](#).

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 \(p. 371\)](#).

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.

The screenshot shows the AWS Management Console interface for monitoring EBS volume events. At the top, there are filters for 'Volume resources', 'All event types', and 'Ongoing and scheduled'. A search bar is present. Below the filters is a table with columns: Resource Name, Resource Type, Resource Id, Availability Zone, Event Type, Event Description, Event Status, Start Time, and Duration. Two rows are visible, both for 'volume' resources in 'us-east-1d' availability zone. The second row is highlighted, showing a 'potential-data-inconsistency' event with a status of 'Awaiting Action: Enable IO'. Below the table, a detailed view for the event 'vol-3682c675' is shown, including a warning icon and a message: 'IO operations have been disabled since 30 days, 15 hours and 22 minutes ago. Data inconsistencies may exist.' An 'Enable Volume IO' button is visible. The detailed view lists: Availability Zone: us-east-1d, Event Type: potential-data-inconsistency, Event Status: Awaiting Action: Enable IO, IO status: IO Disabled, Attached to: i-93aae4ea, Start Time: December 23, 2013 7:09:20 PM UTC+2, and End time. A link to 'Find out more about monitoring volume events.' is at the bottom.

If you have a volume where I/O is disabled, see [Working with an Impaired Volume \(p. 371\)](#). 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 \(p. 371\)](#)
- [Option 2: Perform a Consistency Check on the Volume Using Another Instance \(p. 372\)](#)
- [Option 3: Delete the Volume If You No Longer Need It \(p. 373\)](#)

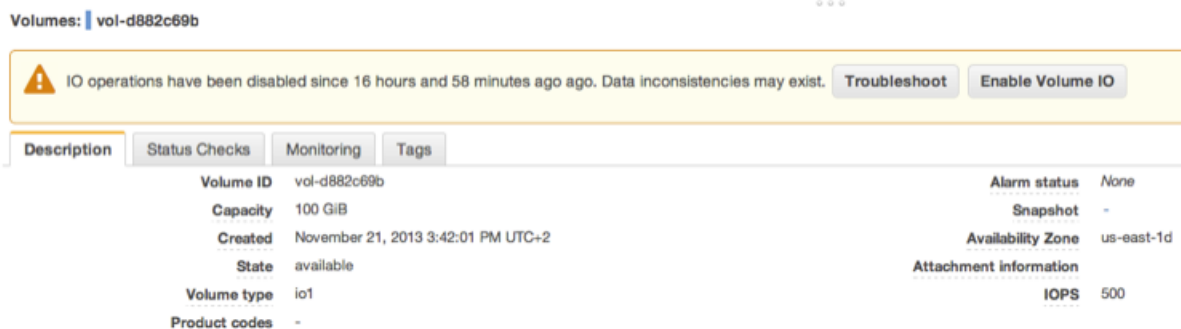
Option 1: Perform a Consistency Check on the Volume Attached to its Instance

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 **chkdsk** 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 Elastic Compute Cloud Command Line Reference*. For information about using the API to enable I/O for a volume, see [EnableVolumeIO](#) in the *Amazon Elastic Compute Cloud 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.

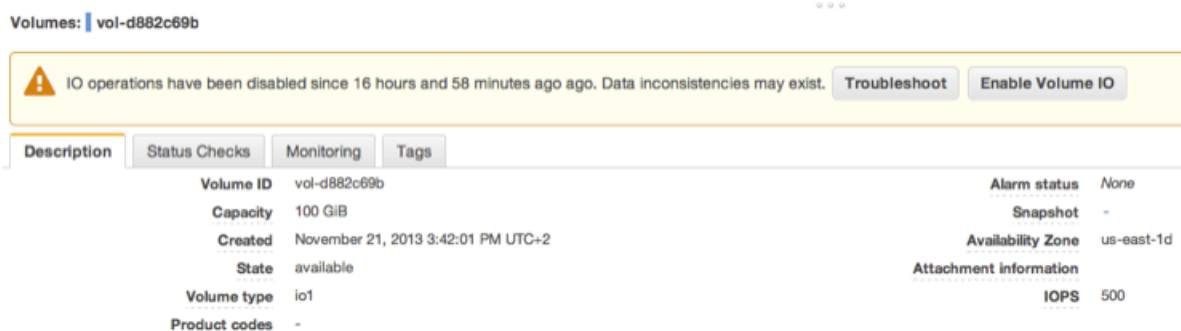
Important

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 \(p. 129\)](#) and [Attaching an Amazon EBS Volume to an Instance \(p. 361\)](#).
5. Check the data on the volume.
 - a. Run the **chkdsk** 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 Elastic Compute Cloud Command Line Reference*. For information about using the API to enable I/O for a volume, see [EnableVolumeIO](#) in the *Amazon Elastic Compute Cloud 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 \(p. 376\)](#).

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 \(p. 359\)](#).

Working with the AutoEnableIO Volume Attribute

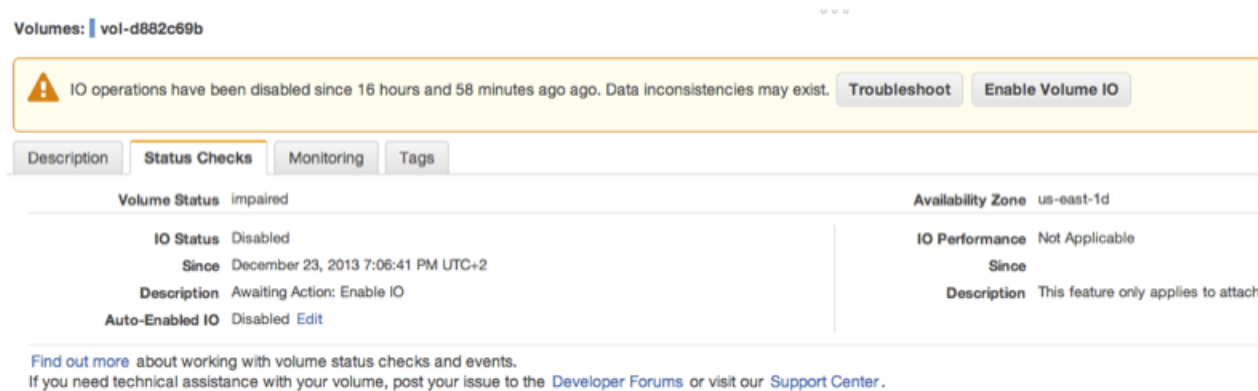
When Amazon EBS determines that a volume's data is potentially inconsistent, it disables I/O to the volume from any attached EC2 instances by default. This causes the volume status check to fail, and creates a volume status event that indicates the cause of the failure. If the consistency of a particular 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. 370).

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.

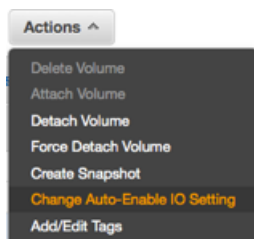
To view the `AutoEnableIO` attribute of a volume in the console

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`.

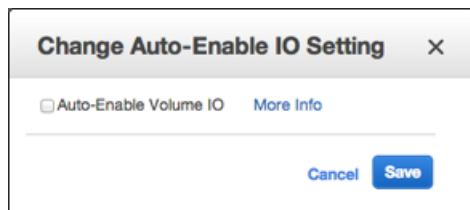


To modify the `AutoEnableIO` attribute of a volume in the console

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 Amazon EBS volume is the root device of an instance, you must stop the instance before you can detach the volume.

When a root 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 \(p. 376\)](#).

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 \(p. 365\)](#).

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 Amazon 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 Amazon 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 a 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 a 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 Amazon 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 a Volume

Sometimes, it is necessary for you to increase the storage space of an existing volume without losing the data that is on the volume. This topic explains how to expand the storage space of an Amazon EBS volume by migrating your data to a larger volume, and then extending the file system on the volume to recognize the newly-available space. After you verify that your new volume is working properly, you may delete the old volume.

Topics

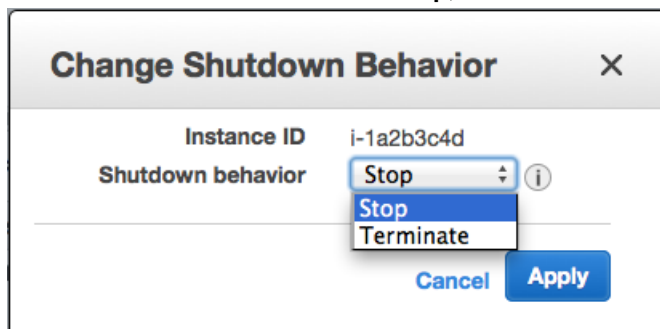
- [Migrating Your Data to a Larger Volume](#) (p. 377)
- [Extending a Windows File System](#) (p. 378)
- [Deleting the Old Volume](#) (p. 381)

Migrating Your Data to a Larger Volume

To migrate your data to a larger Amazon EBS volume

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Ensure that the instance's **Shutdown Behavior** value is set to **Stop** and not **Terminate**. If it is already set to **Stop**, go on to step 3.
 - a. In the navigation pane, click **Instances**, right-click on the instance to check, and select **Change Shutdown Behavior**.
 - b. If the **Shutdown behavior** is set to **Terminate**, select **Stop** from the list and click **Apply**.

If the **Shutdown behavior** is set to **Stop**, click **Cancel**.



3. Stop the instance. For more information about how to stop an instance, see [Stopping and Starting Your Instances](#) (p. 141).
4. Create a snapshot of the volume to expand.
 - a. In the navigation pane, click **Volumes**, right-click on the volume to be expanded, and select **Create Snapshot**.
 - b. Enter a **Name** and **Description** for the snapshot, and click **Yes, Create**.
5. Create a new volume from the snapshot.
 - a. In the navigation pane, click **Snapshots**.
 - b. When the status of the snapshot that you just created is set to **completed**, select the snapshot and click **Create Volume**.

Note

It can take several minutes for the snapshot to complete.

- c. In the **Create Volume** dialog box, select the desired volume type, enter the new size for the volume, set the Availability Zone to the same Availability Zone as the instance, and click **Yes, Create**.
6. Detach the old volume.
 - a. In the navigation pane, click **Volumes**, select the old volume from the list of volumes, and make note of the value of *device name* in **Attachment information**. The attachment information value takes the following form:

```
instance information:device name
```
 - b. Right-click the old volume and select **Detach Volume**.
 - c. In the **Detach Volume** dialog box, click **Yes, Detach**. It may take several minutes for the volume to be detached.
 7. Attach the newly expanded volume
 - a. In the navigation pane, click **Volumes**, select the new volume from the list of volumes, right-click the new volume, and select **Attach Volume**.
 - b. Start typing the name or ID of the instance in the **Instance** field, select the instance, enter the same device name retrieved in [Step 6.a \(p. 378\)](#), and click **Yes, Attach**.
 8. Restart the instance.
 - a. In the navigation pane, click **Instances**, right-click the instance, and select **Start**.
 - b. In the **Start Instances** dialog box, select **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 (root volumes must be attached as `/dev/sda1`).

Important

Only EC2-VPC instances with Elastic IP addresses retain their public IP address when they are stopped. If your instance is running in EC2-Classic, the EIP address is disassociated when the instance is stopped, and you must re-associate the EIP after restarting the instance. For more information, see [Elastic IP Addresses \(EIP\) \(p. 329\)](#). If your instance is not using an EIP, then you need to retrieve your new public DNS name for your instance from the Instances page of the Amazon EC2 console to connect to it.

After the instance has started, you can check the file system size to see if your instance recognizes the larger volume space.

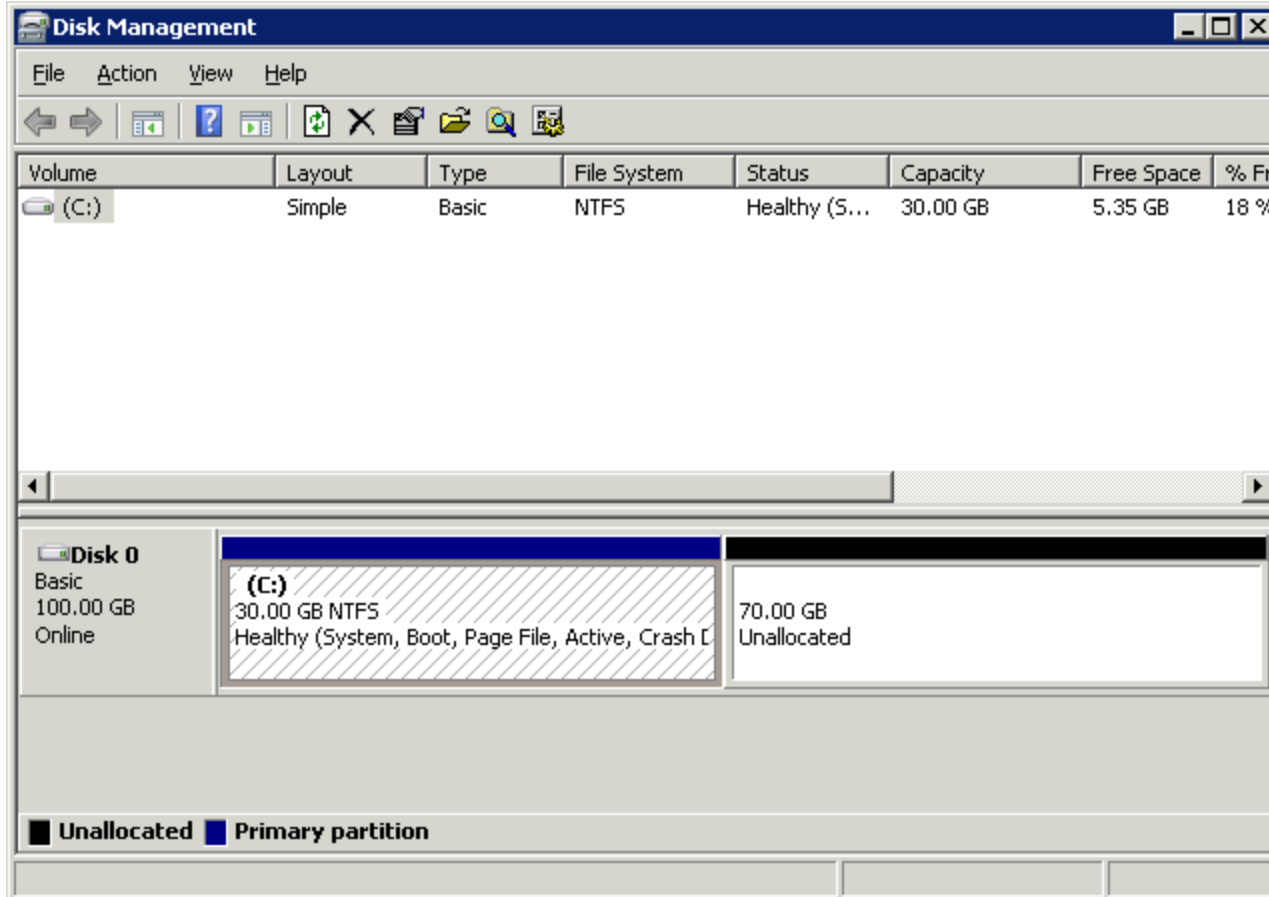
If the size does not reflect your newly-expanded volume, you must extend the file system your device so that your instance can use the new space. For more information, see [Extending a Windows File System \(p. 378\)](#).

Extending a Windows File System

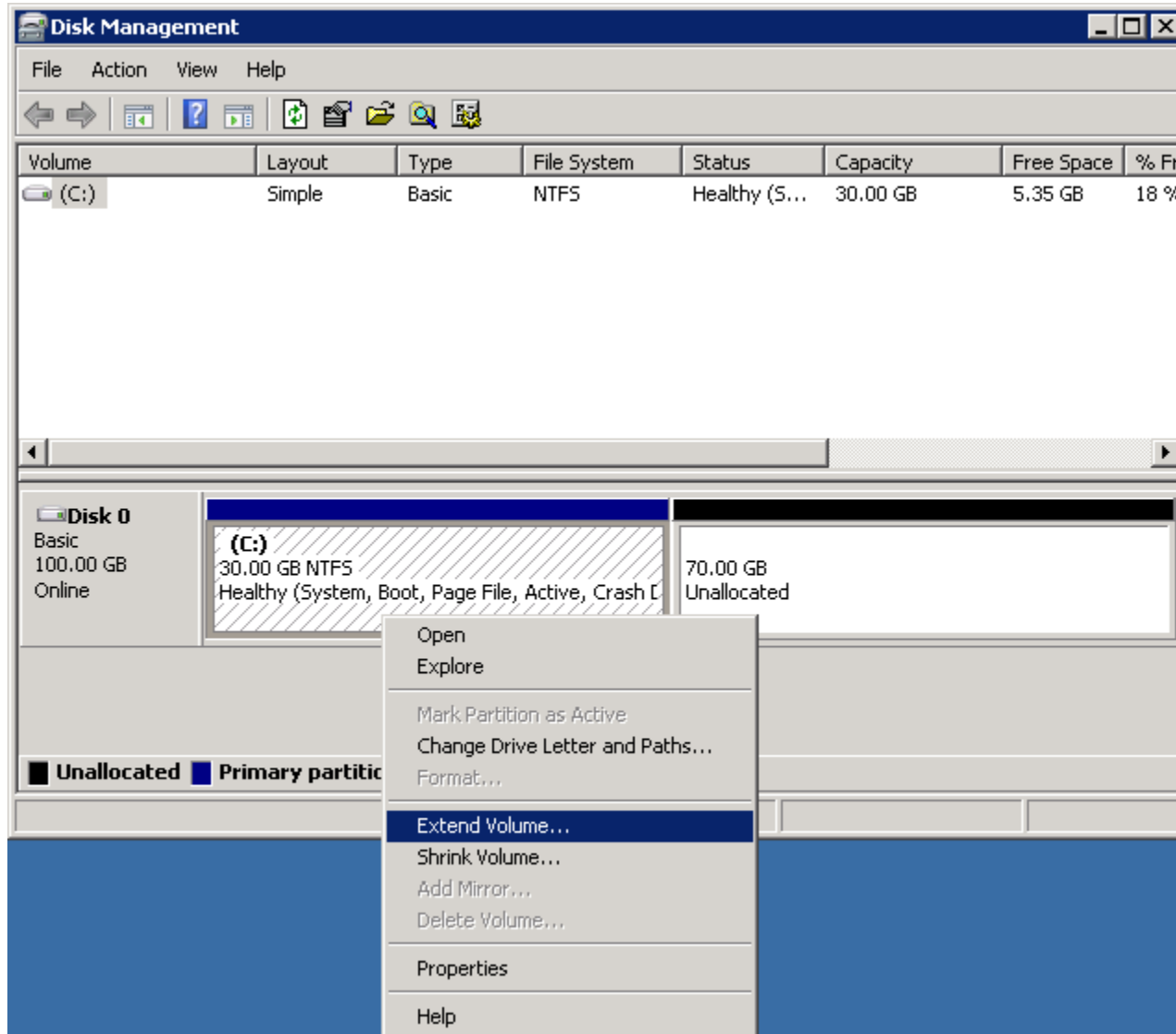
In Windows, you use the Disk Management utility to extend the disk size to the new size of the volume.

To extend a Windows file system

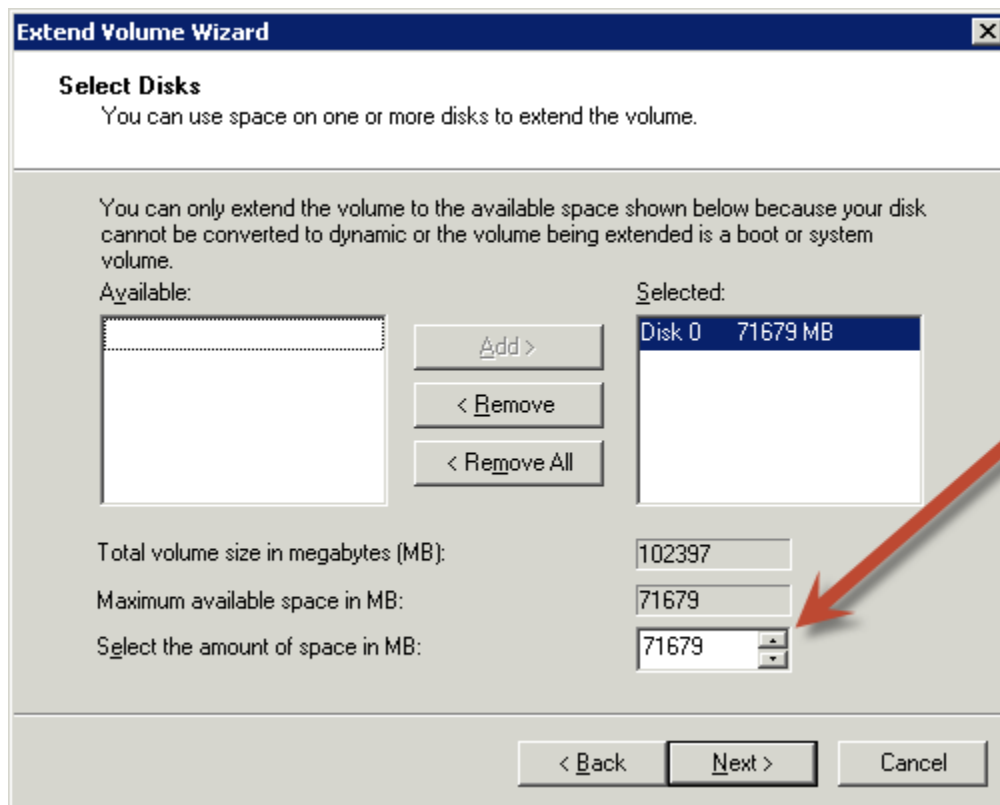
1. Log in to your Windows instance using Remote Desktop.
2.
 - Windows Server 2012: Go to the Start screen.
 - Windows Server 2008: On the taskbar, click **Start**, and then click **Run**.
3. Type **diskmgmt.msc** and press **Enter**. The **Disk Management** utility opens.



4. Right-click the expanded drive and select **Extend Volume**.



5. In the Extend Volume Wizard, click **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, click **Volumes** in the navigation pane.
2. Right-click the old volume and select **Delete Volume**.
3. In the **Delete Volume** dialog box, click **Yes, Delete**.

Amazon EBS Snapshots

An Amazon EBS snapshot is a point-in-time backup copy of an Amazon EBS volume that is stored in Amazon S3. 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 Amazon 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 \(p. 387\)](#).

Contents

- [Creating an Amazon EBS Snapshot \(p. 382\)](#)
- [Deleting an Amazon EBS Snapshot \(p. 383\)](#)

- [Copying an Amazon EBS Snapshot \(p. 384\)](#)
- [Viewing Snapshot Information \(p. 385\)](#)
- [Sharing Snapshots \(p. 386\)](#)

When you create a new Amazon 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. New volumes created from existing Amazon S3 snapshots load lazily in the background, so you can begin using them right away. If your instance accesses a piece of data that hasn't yet been loaded, 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 about creating snapshots, see [Creating an Amazon EBS Snapshot \(p. 382\)](#).

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 \(p. 387\)](#).

You can share your unencrypted snapshots with specific individuals, or make them public to share them with the entire AWS community. Users with access to your snapshots can create their own Amazon EBS volumes from your snapshot, but your snapshots remain completely intact. For more information about how to share snapshots, see [Sharing Snapshots \(p. 386\)](#). Encrypted snapshots cannot be shared with anyone, because your volume encryption keys and master key are specific to your account. If you need to share your encrypted snapshot data, you can migrate the data to an unencrypted volume and share a snapshot of that volume. For more information, see [Migrating Data \(p. 388\)](#).

Amazon EBS snapshots are constrained to the region in which they are created. Once you have created a snapshot of an Amazon 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 \(p. 359\)](#). You can also copy snapshots across AWS regions, making it easier to leverage multiple AWS regions for geographical expansion, data center migration and disaster recovery. You can copy any accessible snapshots that are in the `available` state. For more information, see [Copying an Amazon EBS Snapshot \(p. 384\)](#).

Creating an Amazon EBS Snapshot

After writing data to an Amazon 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 and the status of the snapshot is `pending` until the snapshot is complete.

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 \(p. 387\)](#).

By default, only you can launch 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 Snapshots \(p. 386\)](#). Encrypted snapshots cannot be shared with anyone, because your volume encryption keys and master key are specific to your account. If you need to share your encrypted snapshot data, you can migrate the data to an unencrypted volume and share a snapshot of that volume. For more information, see [Migrating Data \(p. 388\)](#).

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.
2. Click **Snapshots** in the navigation pane.
3. Click **Create Snapshot**.
4. In the **Create Snapshot** dialog box, select the volume to create a snapshot for, and then click **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

This section describes how to delete a snapshot.

Note

- 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.
- You cannot delete a snapshot of the root device of an Amazon 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 \(p. 72\)](#).

To delete a snapshot using the console

1. Open the Amazon EC2 console.
2. Click **Snapshots** in the navigation pane.
3. Select a snapshot and then select **Delete** from the **Actions** list.
4. Click **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

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with Amazon Elastic Compute Cloud (Amazon EC2) instances. With Amazon EBS, you can create point-in-time snapshots of volumes and store them on Amazon Simple Storage Service (Amazon S3). After you've stored a snapshot in Amazon S3, you can copy it from one AWS region to another, or within the same region, using the Amazon EC2 console, Amazon EC2 CLI, or the API. You can use snapshot copies to create backups of data, to create new Amazon EBS volumes, or to create Amazon Machine Images (AMIs).

You can have up to five snapshot copy requests in progress to a single destination per account. You can copy any accessible Amazon EBS snapshots that have "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.

The first snapshot copy of a volume is always a full copy. Each subsequent snapshot copy is incremental, meaning that only the blocks on the volume that have changed since your last snapshot copy to the same destination are transferred. Incremental snapshots make the copy process faster. Support for incremental snapshots is specific to a region pair. For example, if you copy a snapshot from the US East (Northern Virginia) Region to the US West (Oregon) Region, the first snapshot copy of the volume is a full copy. However, subsequent snapshot copies of the same volume transferred between the same regions are incremental. A snapshot copy can only be done incrementally as long as there is one full copy of the same volume available in the destination region.

Note

To copy an Amazon Relational Database Service (Amazon RDS) snapshot, see [Copying a DB Snapshot](#) in the Amazon Relational Database Service User Guide.

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 snapshot is copied across regions using the secure Amazon S3 Copy and the snapshot copy receives a snapshot ID that's different from the original snapshot's ID.

You can use a copy of an Amazon EBS snapshot in the following ways:

- **Geographic Expansion:** You can launch your applications in a new region.
- **Migration:** You can migrate an application to a new region, to enable better availability and minimize cost.
- **Disaster Recovery:** You can 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.

The Amazon EC2 console, Amazon EC2 CLI, and the API are designed to provide an intuitive customer experience. We use the push model in the console design to minimize user clicks for the Amazon EBS snapshot use cases discussed earlier. You can easily initiate a copy from the console by starting with the source region. We use a pull model in the Amazon EC2 CLI and the API, because these experiences

factor in how customers use automation. You only need to know the source snapshot ID and source region to initiate the copy using the Amazon EC2 CLI or API.

To copy a snapshot using the Amazon EC2 console

You can create a copy of an Amazon EBS snapshot using the Amazon EC2 console.

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Snapshots**.
3. Select the snapshot to copy, and then select **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.
5. Click **Yes, Copy**.
6. In the **Copy Snapshot** confirmation dialog box, you can click **Snapshots** to go to the **Snapshots** page in the region specified, or click **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 Snapshot Information

This section describes how to view descriptive information about the snapshots that you have created.

To view descriptive snapshot information in the console

1. Open the Amazon EC2 console.
2. Click **Snapshots** in the navigation pane.
3. To reduce the list, select an option from the **Filter** list. For example, to view only your snapshots, select **Owned By Me**.
4. To view more information about a snapshot, select it.

To view descriptive 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 Snapshots

This section describes how to share your unencrypted Amazon EBS snapshots with your co-workers or others in the AWS community by modifying the permissions of the snapshot. Encrypted snapshots cannot be shared with anyone, because the keys that are used to encrypt your volumes and snapshots are specific to your account. If you need to share your encrypted snapshot data, you can migrate the data to an unencrypted volume and share a snapshot of that volume. For more information, see [Migrating Data](#) (p. 388).

Users that you have authorized can quickly use your unencrypted Amazon EBS shared snapshots as the basis for creating their own Amazon EBS volumes. If you choose, you can also make your data available publicly to all AWS users. Users to whom you have granted access can create their own Amazon EBS volumes based on your snapshot and your original snapshot remains intact.

Note

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. 384).

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 in the console

This procedure will help you to share your unencrypted snapshots. For security reasons, encrypted snapshots cannot be shared or made public.

1. Open the Amazon EC2 console.
2. Click **Snapshots** in the navigation pane.
3. Select a snapshot and then select **Modify Snapshot Permissions** from the **Actions** list.
4. Choose whether to make the snapshot public or to share it with select AWS accounts:

Important

Making your snapshot public shares all snapshot data with everyone. Snapshots with AWS Marketplace product codes cannot be made public.

- To make the snapshot public, select **Public**.
- To expose the snapshot only to specific AWS accounts, select **Private**, enter the ID of the AWS account (without hyphens) in the **AWS Account Number** field, and click **Add Permission**. Repeat until you've added all the required AWS accounts.

Click **Save** when you're done.

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 Encryption

Amazon EBS encryption offers you a simple encryption solution for your Amazon 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 Amazon EC2 instances, providing encryption of data-in-transit from EC2 instances to EBS storage.

This feature is supported on all Amazon EBS volume types (General Purpose (SSD), Provisioned IOPS (SSD), and Magnetic), and you can expect the same provisioned IOPS performance on encrypted volumes as you would with unencrypted volumes with a minimal effect on latency. You can access encrypted Amazon EBS volumes the same way you access existing volumes; encryption and decryption are handled transparently and they require no additional action from you, your EC2 instance, or your application. Snapshots of encrypted EBS volumes are automatically encrypted, and volumes that are created from encrypted EBS snapshots are also automatically encrypted.

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. For more information, see [Supported Instance Types \(p. 387\)](#). You can attach both encrypted and unencrypted volumes to a supported instance type. You can use and manage encrypted EBS volumes and snapshots using the AWS Management Console, command line interface (CLI), AWS SDKs, or the Amazon EC2 API.

Topics

- [Encryption Key Management \(p. 387\)](#)
- [Supported Instance Types \(p. 387\)](#)
- [Considerations \(p. 388\)](#)
- [Migrating Data \(p. 388\)](#)

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.

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
Compute optimized	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	i2.xlarge i2.2xlarge i2.4xlarge i2.8xlarge
GPU instances	g2.2xlarge

For more information on 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 or vice versa.

Public or shared snapshots of encrypted volumes are not supported, because other accounts would not 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 \(p. 388\)](#).

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. 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.

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 \(p. 357\)](#).
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 \(p. 361\)](#).
3. Make the destination volume available by following the procedures in [Making an Amazon EBS Volume Available for Use \(p. 363\)](#).
4. Copy the data from your source directory to the destination volume.

From the Command Prompt window, 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

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.

Topics

- [Amazon EBS Performance Tips \(p. 389\)](#)
- [Amazon EC2 Instance Configuration \(p. 390\)](#)
- [I/O Characteristics \(p. 391\)](#)
- [Workload Demand \(p. 392\)](#)
- [Pre-Warming Amazon EBS Volumes \(p. 392\)](#)
- [RAID Configuration \(p. 394\)](#)
- [Benchmark Volumes \(p. 399\)](#)

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 \(p. 390\)](#).
- 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 \(p. 391\)](#).
- 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 \(p. 392\)](#).
- There is a 5 to 50 percent reduction in IOPS 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 \(p. 392\)](#).
- General Purpose (SSD) and Provisioned IOPS (SSD) volumes have a throughput limit of 128 MB/s per volume. Some instance types can drive more I/O throughput than you can provision for a single 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 \(p. 394\)](#).
- 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 \(p. 399\)](#).
- 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 \(p. 365\)](#).
- 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 \(p. 381\)](#).

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	Dedicated EBS 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
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

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
EBS Performance**

Instance Type	Dedicated EBS Throughput (Mbps)*	Max 16K IOPS**	Max Bandwidth (MB/s)**
m3.xlarge	500	4,000	62.5
m3.2xlarge	1,000	8,000	125
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 two 4,000 IOPS volumes are attached to an EBS-optimized `m1.large` instance in a RAID 0 configuration, the EC2 to EBS connection bandwidth limit will prevent these volumes from providing the 256 MB/s maximum aggregate throughput available to them. In this case, we must use an EBS-optimized EC2 instance that supports 250 MB/s of throughput, such as the `r3.4xlarge` instance type.

General Purpose (SSD) and Provisioned IOPS (SSD) volumes have a throughput limit of 128 MB/s per volume, which pairs nicely 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) or Provisioned IOPS (SSD) volume to take advantage of the available throughput.

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. Since the maximum provisioned IOPS value for EBS volumes is 4,000, you can use many EBS volumes simultaneously to reach the level of I/O performance available to these instance types. To see 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 \(p. 93\)](#).

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 KB or smaller) as one IOPS. I/O operations that are larger than 256 KB are counted in 256 KB capacity units. For example, a 1,024 KB I/O operation would count as 4 IOPS. When you provision a 4,000 IOPS volume and attach it to an EBS-optimized instance that can provide the necessary bandwidth, you can transfer up to 4,000 chunks of data per second (provided that the I/O does not exceed the 128 MB/s per volume throughput limit of General Purpose (SSD) and Provisioned IOPS (SSD) volumes).

This configuration could transfer 4,000 32 KB chunks, 2,000 64 KB chunks, or 1,000 128 KB chunks of data per second as well, before hitting the 128 MB/s per volume throughput limit. If your I/O chunks are

very large, you may experience a smaller number of IOPS than you provisioned because you are hitting the volume throughput limit.

For 32 KB or smaller I/O operations, you should see the amount of IOPS that you have provisioned, provided that you are driving enough I/O to keep the drives busy. 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 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 \(p. 390\)](#). 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 \(p. 392\)](#).

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. We recommend that you target an optimal average queue length of 1 for every 200 IOPS provisioned and tune that value based on your application requirements. For example, a volume with 1,000 IOPS provisioned should target an average queue length of 5.

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 (which is achieved by provisioning more IOPS for your volume).

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 5 to 50 percent loss of IOPS for your volume 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.

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=\\.\PHYSICALDRIVE0 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 \(p. 363\)](#).

RAID Configuration

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.

Note

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.

This topic provides basic RAID setup examples. For much more detailed information on RAID configuration, performance, and recovery, see the Linux RAID Wiki at https://raid.wiki.kernel.org/index.php/Linux_Raid.

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 1 TiB RAID 0 array with an available bandwidth of 8,000 IOPS and 256 MB/s of throughput or a 500 GiB RAID 1 array with an available bandwidth of 4,000 IOPS and 128 MB/s of throughput.

Creating a RAID Array on Windows

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 \(p. 357\)](#).

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 \(p. 390\)](#).

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 \(p. 361\)](#).
3. Connect to your Windows instance. For more information, see [Connecting to Your Windows Instance Using RDP \(p. 138\)](#).
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
```

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
EBS Performance**

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 \(p. 396\)](#) 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 2,000 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 \(p. 93\)](#).

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 \(p. 361\)](#).

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 between a 5 to 50 percent reduction in IOPS when you first 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 \(p. 376\)](#) and [Terminate Your Instance \(p. 145\)](#).

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
```

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
```

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:
```

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
API and Command Overview**

```
\\.\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 API and Command Overview

The following table summarizes the available commands for Amazon EBS and corresponding API actions for creating and using Amazon EBS volumes.

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 <code>running</code> or <code>stopped</code> state.
copy-snapshot (AWS CLI) ec2-copy-snapshot (Amazon EC2 CLI) CopySnapshot	Copies a point-in-time snapshot of an Amazon 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 Amazon EBS volumes or AMIs.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
API and Command Overview**

Command/Action	Description
create-snapshot (AWS CLI) ec2-create-snapshot (Amazon EC2 CLI) CreateSnapshot	<p>Creates a snapshot of the volume you specify.</p> <p>After the snapshot is created, you can use it to create volumes that contain exactly the same data as the original volume.</p>
create-volume (AWS CLI) ec2-create-volume (Amazon EC2 CLI) CreateVolume	<p>Creates a new Amazon EBS volume using the specified size and type, or based on a previously created snapshot.</p>
delete-snapshot (AWS CLI) ec2-delete-snapshot (Amazon EC2 CLI) DeleteSnapshot	<p>Deletes the specified snapshot.</p> <p>This command does not affect currently running Amazon EBS volumes, regardless of whether they were used to create the snapshot or were derived from the snapshot.</p>
delete-volume (AWS CLI) ec2-delete-volume (Amazon EC2 CLI) DeleteVolume	<p>Deletes the specified volume. The command does not delete any snapshots that were created from the volume.</p>
describe-snapshot-attribute (AWS CLI) ec2-describe-snapshot-attribute (Amazon EC2 CLI) DescribeSnapshotAttribute	<p>Describes attributes for a snapshot.</p>
describe-snapshots (AWS CLI) ec2-describe-snapshots (Amazon EC2 CLI) DescribeSnapshots	<p>Describes the specified snapshot.</p> <p>Describes all snapshots, including their source volume, snapshot initiation time, progress (percentage complete), and status (<i>pending</i>, <i>completed</i>, and so on.).</p>
describe-volume-attribute (AWS CLI) ec2-describe-volume-attribute (Amazon EC2 CLI) DescribeVolumeAttribute	<p>Describes an attribute of a volume.</p>
describe-volume-status (AWS CLI) ec2-describe-volume-status (Amazon EC2 CLI) DescribeVolumeStatus	<p>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.</p>

Command/Action	Description
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 (<i>available</i> or <i>in-use</i>). If the volume is <i>in-use</i> , 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 (<i>attaching</i> , <i>attached</i> , <i>detaching</i> , <i>detached</i>), 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 <code>all</code> 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.
reset-snapshot-attribute (AWS CLI) ec2-reset-snapshot-attribute (Amazon EC2 CLI) ResetSnapshotAttribute	Resets permission settings for the specified snapshot.

Amazon EC2 Instance Store

Many Amazon EC2 instance types can access disk storage from disks that are physically attached to the host computer. This disk storage is referred to as *instance store*.

Contents

- [Instance Storage Concepts](#) (p. 404)
- [Instance Stores Available on Instance Types](#) (p. 405)
- [Instance Store Device Names](#) (p. 406)
- [Instance Store Usage Scenarios](#) (p. 407)

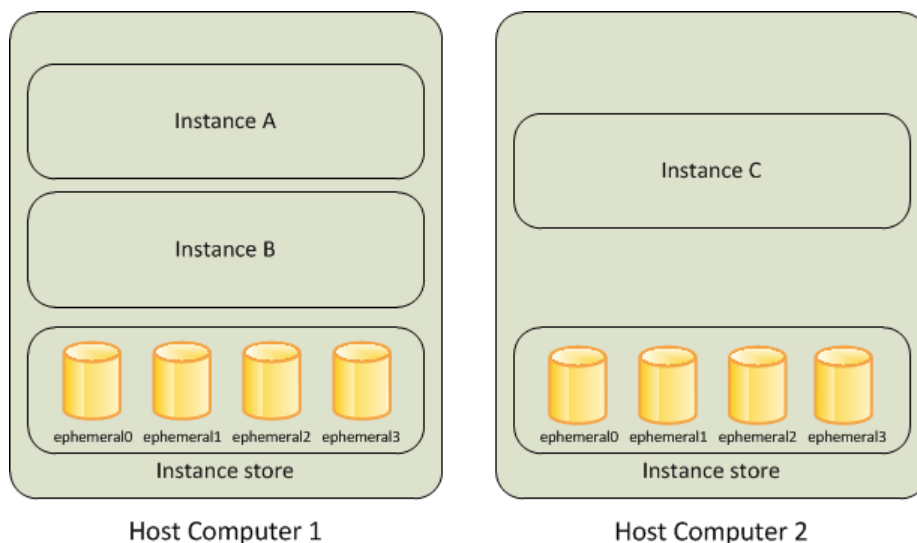
- [Adding Instance Store Volumes to an AMI \(p. 408\)](#)

Instance Storage Concepts

An instance store provides temporary block-level storage for use with an instance. The size of an instance store ranges from 900 MiB to up to 48 TiB, and varies by instance type. Larger instance types have larger instance stores. Some smaller instance families, such as T2 and T1, do not support instance store volumes at all and they use Amazon EBS exclusively for storage. For more information, see [Instance Stores Available on Instance Types \(p. 405\)](#).

An instance store consists of one or more instance store volumes. When you launch an instance store-backed AMI, each instance store volume available to the instance is automatically mapped. When you launch an Amazon EBS-backed AMI, instance store volumes must be configured using block device mapping at launch time (with either the default block device mapping for the chosen AMI or manually using the console or the CLI or SDK tools). Volumes must be formatted and mounted on the running instance before they can be used. By default, instances launched from an Amazon EBS-backed AMI have no mounted instance store volumes. Instances launched from an instance store-backed AMI have a mounted instance store volume for the virtual machine's root device volume (the size of this volume varies by AMI, but the maximum size is 10 GiB) in addition to the instance store volumes included with the instance type. For more information about instance store-backed AMIs and Amazon EBS-backed AMIs, see [Storage for the Root Device \(p. 48\)](#).

Instance store volumes are usable only from a single instance during its lifetime; they can't be detached and then attached to another 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 for the instances that you launch from this AMI. While an instance store is dedicated to a particular instance, the disk subsystem is shared among instances on a host computer, as shown in the following figure.



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 on instance store volumes is lost under the following circumstances:

- Failure of an underlying drive
- Stopping an Amazon EBS-backed instance
- Terminating an instance

Therefore, do not rely on instance store volumes for valuable, long-term data. Instead, keep your data safe by using a replication strategy across multiple instances, storing data in Amazon S3, or using Amazon EBS volumes. For more information, see [Amazon Elastic Block Store \(Amazon EBS\)](#) (p. 351).

When you launch an instance, whether it's launched from an Amazon EBS-backed AMI or an instance store-backed AMI, you can attach instance store volumes to the instance using block device mapping. For more information, see [Adding Instance Store Volumes to an AMI](#) (p. 408).

Instance Stores Available on Instance Types

Amazon EC2 instances are divided into different instance types, which determine the size of the instance store available on the instance by default. When you launch an instance, you can specify an instance type or use the default instance type, which is an `m1.small` instance.

The instance type also determines the type of hardware for your 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 it to persist when the instance terminates, or you can take advantage of fault tolerant architectures. For more information see [H1 Instances](#) (p. 81).

The following table shows the instance types along with the size and quantity of the instance store volumes available to each instance type; these instance store volumes are included as part of the instance's hourly cost.

Instance Type	Instance Store Volumes
<code>c1.medium</code>	1 x 350 GB
<code>c1.xlarge</code>	4 x 420 GB (1680 GB)
<code>c3.large</code>	2 x 16 GB SSD (32 GB)
<code>c3.xlarge</code>	2 x 40 GB SSD (80 GB)
<code>c3.2xlarge</code>	2 x 80 GB SSD (160 GB)
<code>c3.4xlarge</code>	2 x 160 GB SSD (320 GB)
<code>c3.8xlarge</code>	2 x 320 GB SSD (640 GB)
<code>cc2.8xlarge</code>	4 x 840 GB (3360 GB)
<code>cgl.4xlarge</code>	2 x 840 GB (1680 GB)
<code>cr1.8xlarge</code>	2 x 120 GB SSD (240 GB)
<code>g2.2xlarge</code>	1 x 60 GB SSD
<code>hi1.4xlarge</code>	2 x 1024 GB SSD (2048 GB)
<code>hs1.8xlarge</code>	24 x 2048 GB (49 TB)
<code>i2.xlarge</code>	1 x 800 GB SSD
<code>i2.2xlarge</code>	2 x 800 GB SSD (1600 GB)
<code>i2.4xlarge</code>	4 x 800 GB SSD (3200 GB)
<code>i2.8xlarge</code>	8 x 800 GB SSD (6400 GB)
<code>m1.small</code>	1 x 160 GB

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Instance Store Device Names**

Instance Type	Instance Store Volumes
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
r3.xlarge	1 x 80 GB
r3.2xlarge	1 x 160 GB
r3.4xlarge	1 x 320 GB
r3.8xlarge	2 X 320 GB (640 GB)
t1.micro	None (use Amazon EBS volumes)
t2.micro	None (use Amazon EBS volumes)
t2.small	None (use Amazon EBS volumes)
t2.medium	None (use Amazon EBS volumes)

Instance Store Device Names

Within an instance store, instance store volumes are exposed as block devices. 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`. Instance types that support four instance store volumes have `ephemeral0`, `ephemeral1`, `ephemeral2`, and `ephemeral3`, and so on.

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. A Windows instance uses a built-in tool, EC2Config Service, to reformat the instance store volumes available on an instance with the NTFS file system.

Each entry in a block device mapping consists of a device name and the volume that it's mapped to. The instance store volumes are available to the instance, but you can't access them until they are mounted. A Windows instance uses the EC2Config Service to mount 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.

An instance can have multiple instance store volumes mapped to a device. However, the number and size of these volumes must not exceed the instance store available for the instance type. For more information, see [Instance Stores Available on Instance Types \(p. 405\)](#).

Instance Store Usage Scenarios

Instance store volumes are 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.

Making Instance Stores Available on Your Instances

Instances that use Amazon EBS for the root device do not, by default, have instance store available at boot time. Also, you can't attach instance store volumes after you've launched an instance. Therefore, if you want your Amazon EBS-backed instance to use instance store volumes, you must specify them using a block device mapping when you create your AMI or launch your instance. Examples of block device mapping entries are: `/dev/sdb=ephemeral0` and `/dev/sdc=ephemeral1`. For more information about block device mapping, see [Block Device Mapping \(p. 411\)](#)

The following procedure describes how to launch an Amazon EBS-backed `m1.large` Windows instance with instance store volumes.

Launch Amazon EBS-backed Windows Instances with Instance Store Volumes

1. Locate an Amazon EBS-backed Windows AMI.
2. Launch an instance that supports at least two instance store volumes with this AMI and add block device mapping entries for `ephemeral0` and `ephemeral1`.

For more information, see [To add volumes to an instance \(p. 416\)](#).

3. Connect to the instance.
4. On the **Start** menu, choose **Computer**.
5. Devices listed:
 - Local Disk C:/ 9.98GiB
 - Local Disk D:/ 419GiB
 - Local Disk E:/ 419GiB
6. Double-click **Local Disk C:/**. You can see the list of installed applications. This is your root drive.
7. Double-click **Local Disk D:/** and then double-click **Local Disk E:/**. These drives are empty. They are the instance stores that come with your `m1.large` instance, and they are available for you to use with your applications.

You can also map instance store volumes to block devices when you create an AMI. The instances launched from such an AMI have instance store volumes at boot time. For information about adding a block device mapping while creating an AMI, see [Creating an Amazon EBS-Backed Windows AMI \(p. 62\)](#).

The following procedure describes how to access the instance store volumes from within an Amazon EC2 instance store-backed `m1.large` Windows instance.

Tasks for Accessing Instance Stores on Amazon EC2 instance store-backed Windows Instances

- | | |
|---|---|
| 1 | Locate an Amazon EC2 instance store-backed Windows AMI. |
|---|---|

2	Launch an <code>m1.large</code> instance.
3	Connect to the instance.
4	On the Start menu, choose Computer .
5	Devices listed: <ul style="list-style-type: none">• Local Disk C:/ 9.98GiB• Local Disk D:/ 419GiB• Local Disk E:/ 419GiB
6	Double-click Local Disk C:/. You see the list of all installed applications. This is your root drive.
7	Double-click Local Disk D:/ and then double-click Local Disk E:/. These are empty. They are the instance store volumes that come with your <code>m1.large</code> instance, and they are available to use with your applications just like any physical drive.

Suppressing Instance Stores at Launch Time

You can prevent a particular instance storage volume from attaching to the instance. You can do this for both Amazon EC2 instance store-backed instances and Amazon EBS-backed instances. For example, specifying the mapping `/dev/sdc=none` when launching an instance prevents `/dev/sdc` from attaching to the instance. For more information about block device mapping, see [Block Device Mapping \(p. 411\)](#).

Adding Instance Store Volumes to an AMI

Amazon EBS-backed AMIs don't include an instance store by default. However, you might want instances launched from your Amazon EBS-backed AMIs to include instance store volumes.

After you add instance store volumes to an AMI, any instance you launch from the AMI includes these instance store volumes. 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 \(p. 420\)](#).

For M3 instances, you must specify instance store volumes in the block device mapping for the instance. When you launch an M3 instance, we ignore any instance store volumes specified in the block device mapping for the AMI.

You can create an AMI that includes instance store volumes using the console or the command line.

To add instance store volumes to an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**.
3. Select an instance and select **Create Image** from the **Actions** list.
4. In the **Create Image** dialog, add a meaningful name and description to your image.
5. For each instance store volume, Click **Add New Volume**, select an instance store volume from the **Type** list and a device name from **Device**.
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)

For more information, see [Block Device Mapping \(p. 411\)](#).

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 \(p. 351\)](#).

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 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_bucket/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.

Block Device Mapping

Each Amazon EC2 instance that you launch has an associated root device volume, either an Amazon Elastic Block Store (Amazon EBS) volume or an instance store volume. You can use block device mapping to specify additional Amazon EBS volumes or instance store volumes to attach to an instance when it's launched. You can also attach additional Amazon EBS volumes to a running instance; see [Attaching an Amazon EBS Volume to an Instance \(p. 361\)](#). 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 \(p. 411\)](#)
- [AMI Block Device Mapping \(p. 414\)](#)
- [Instance Block Device Mapping \(p. 416\)](#)

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)
- Amazon EBS volumes (remote storage devices)

A *block device mapping* defines the block devices to be attached to an Amazon EC2 instance and the device name to use. 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.

There are two types of virtualization available in Amazon EC2: paravirtual (PV) and hardware virtual machine (HVM). The virtualization type is determined by the AMI used to launch the instance; some instance types support both PV and HVM while others support only one or the other. Be sure to note the virtualization type used by your AMI when you are creating your block device mapping because the recommended and available device names that you can use are different based on the virtualization type of your instance.

Specifying a Block Device Mapping

Use a block device mapping to attach instance store volumes and Amazon EBS volumes to an EC2 instance.

When you create a block device mapping, you specify this information for each block device that you need to attach to the instance:

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Block Device Mapping Concepts**

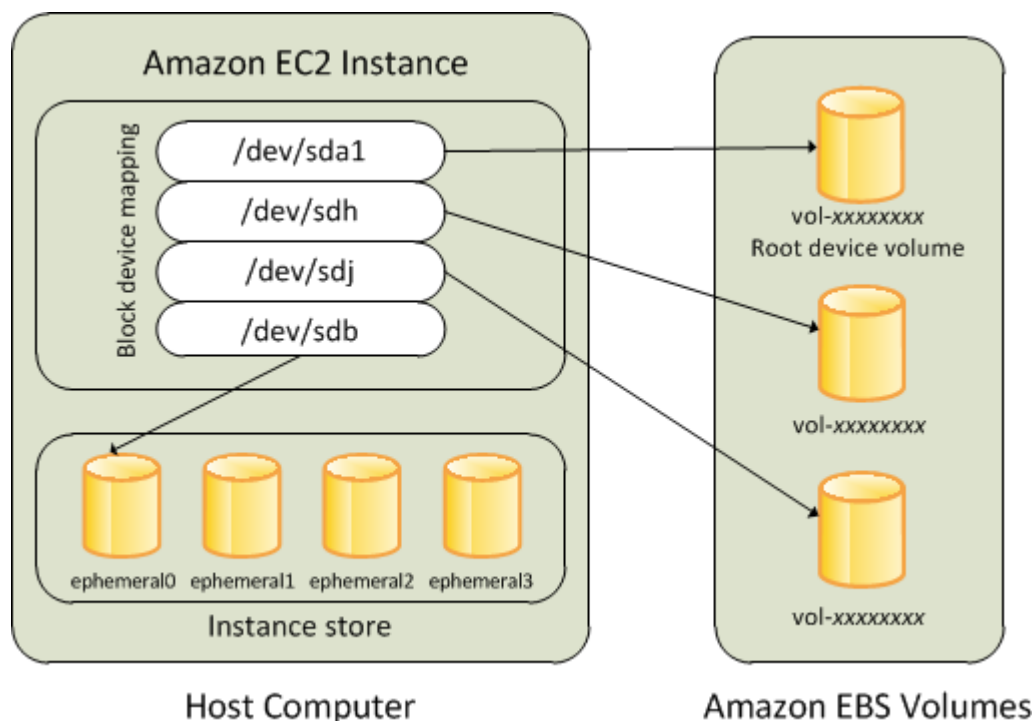
- The device name within Amazon EC2, as shown in this table. The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different from the name that Amazon EC2 recommends.

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]

- [Instance store volumes only] The virtual device: `ephemeral[0-3]`.
- [Amazon EBS volumes only] The ID of the snapshot to use to create the block device (`snap-xxxxxxx`). This value is optional as long as you specify a volume size.
- [Amazon EBS volumes only] The size of the volume, in GiB. The specified size must be greater than or equal to the size of the specified snapshot.
- [Amazon EBS volumes only] Whether to delete the volume on instance termination (`true` or `false`). The default value is `true`.
- [Amazon EBS volumes only] 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 `standard` for Magnetic volumes.
- [Amazon EBS volumes only] The number of input/output operations per second (IOPS) that the volume supports. (Not used with General Purpose (SSD) or Magnetic volumes.)

Example Block Device Mapping

This figure shows an example block device mapping for an Amazon EBS-backed instance. It maps `/dev/sdb` to `ephemeral0` and maps two Amazon EBS volumes, one to `/dev/sdh` and the other to `/dev/sdj`. It also shows the Amazon 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 \(p. 414\)](#)
- [Updating the Block Device Mapping when Launching an Instance \(p. 416\)](#)

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 Amazon 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 Amazon EBS volumes; for more information, see [Configuring a Windows Instance Using the EC2Config Service \(p. 151\)](#).

Viewing Block Device Mappings

You can view information about each block device in a block device mapping. For details, see:

- [Viewing the Amazon EBS Volumes in an AMI Block Device Mapping \(p. 416\)](#)
- [Viewing the Amazon EBS Volumes in an Instance Block Device Mapping \(p. 419\)](#)
- [Viewing the Instance Block Device Mapping for Instance Store Volumes \(p. 420\)](#)

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 \(p. 414\)](#)
- [Viewing the Amazon EBS Volumes in an AMI Block Device Mapping \(p. 416\)](#)

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 Amazon 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 Amazon EBS-backed AMI, you can add Amazon 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.

For M3 instances, you must specify instance store volumes in the block device mapping for the instance. When you launch an M3 instance, we ignore any instance store volumes specified in the block device mapping for the AMI.

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 and select **Create Image** from the **Actions** list.
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 Amazon 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 Amazon 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:

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
AMI Block Device 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 Amazon 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..3]`.

To add an Amazon EBS volume to an Amazon 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 Amazon EBS volume based on a snapshot as `xvdh`, and an empty 100 GiB Amazon EBS volume as `xvdj`.


```
-b "xvdb=ephemeral10" -b "xvdh=snap-d5eb27ab" -b "xvdj=:100"
```

Viewing the Amazon EBS Volumes in an AMI Block Device Mapping

You can easily enumerate the Amazon EBS volumes in the block device mapping for an AMI.

To view the Amazon 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 Amazon 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 Amazon 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 Amazon EBS volumes for an AMI using the AWS CLI

Use the [describe-images](#) command to enumerate the Amazon EBS volumes in the block device mapping for an AMI.

To view the Amazon EBS volumes for an AMI using the Amazon EC2 CLI

Use the [ec2-describe-images](#) command to enumerate the Amazon 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't modify the block device mapping entry for the root device volume.

Contents

- [Updating the Block Device Mapping when Launching an Instance](#) (p. 416)
- [Viewing the Amazon EBS Volumes in an Instance Block Device Mapping](#) (p. 419)
- [Viewing the Instance Block Device Mapping for Instance Store Volumes](#) (p. 420)

Updating the Block Device Mapping when Launching an Instance

You can add Amazon 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 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, you can modify the root volume, Amazon 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 Amazon 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 Amazon 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 Amazon 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  
  }  
}
```

Amazon Elastic Compute Cloud User Guide for Microsoft Windows Instance Block Device Mapping

```
}  
}
```

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..3]`.

To add an Amazon EBS volume to an EBS-backed instance, specify `[snapshot-id]:[size]:[delete-on-termination]:[type[:iops]]`.

- To add an empty Amazon EBS volume, omit the snapshot ID and specify a volume size instead.
- To indicate whether the Amazon 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 Amazon EBS Volumes in an Instance Block Device Mapping

You can easily enumerate the Amazon 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 Amazon EBS volumes for an instance using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**.
3. Select **EBS root device** from the **Filter** list. This displays a list of Amazon 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 Amazon 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.)

Root device type	<i>ebs</i>
Root device	<i>/dev/sda1</i>
Block devices	<i>/dev/sda1</i> <i>/dev/sdf</i>

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-xxxxxxx*)
 - **Root device type** (*ebs*)
 - **Attachment time** (*yyyymmT hh:mm:ss.ssTZD*)
 - **Block device status** (*attaching, attached, detaching, detached*)
 - **Delete on termination** (*Yes, No*)

To view the Amazon EBS volumes for an instance using the AWS CLI

Use the [describe-instances](#) command to enumerate the Amazon EBS volumes in the block device mapping for an instance.

To view the Amazon EBS volumes for an instance using the Amazon EC2 CLI

Use the [ec2-describe-instances](#) command to enumerate the Amazon 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 Amazon 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[0-3]`. 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 \(p. 100\)](#).

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. 421)
- [Finding Public Data Sets](#) (p. 421)
- [Creating a Public Data Set Volume from a Snapshot](#) (p. 422)
- [Attaching and Mounting the Public Data Set Volume](#) (p. 423)

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 Amazon 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 Amazon EBS snapshot format are used to create new Amazon EBS volumes that you attach to an Amazon EC2 instance. For more information, see [Creating a Public Data Set Volume from a Snapshot](#) (p. 422).

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. 409).

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 Amazon EBS 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 Amazon EBS volume to an instance in the same Availability Zone. For more information, see [Resource Locations](#) (p. 424).

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. 384).

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. 355).
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. 421) 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 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

Amazon 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 \(p. 422\)](#)), 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 a Volume \(p. 377\)](#).

Attaching and Mounting the Public Data Set Volume

After you have created your new data set volume, you need to attach it to an Amazon 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 \(p. 361\)](#).

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 \(p. 363\)](#).

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 \(p. 424\)](#)
- [Listing and Filtering Your Resources \(p. 425\)](#)
- [Tagging Your Amazon EC2 Resources \(p. 427\)](#)
- [Amazon EC2 Service Limits \(p. 436\)](#)
- [Amazon EC2 Usage Reports \(p. 437\)](#)

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 (p. 260) .
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.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Listing and Filtering Your Resources**

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. 67) .
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. 384) .
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

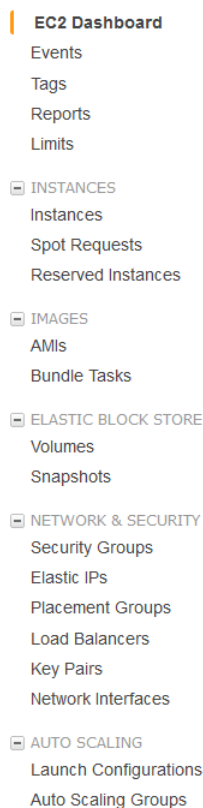
- [Listing Resources Using the Console \(p. 425\)](#)
- [Filtering Resources Using the Console \(p. 426\)](#)
- [Listing and Filtering Using the CLI and API \(p. 427\)](#)

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.

Note

Not all of the screens provide the same search functionality.

To list volumes in the `us-east-1b` Availability Zone with a status of `available`

1. In the navigation pane, click **Volumes**.
2. Select **Detached volumes** from the filter list. (A detached volume is available to be attached to an instance in the same Availability Zone.)

3. Enter `us-east-1b` in the search field.
4. 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.

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).

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 Elastic Compute Cloud 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 Elastic Compute Cloud 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

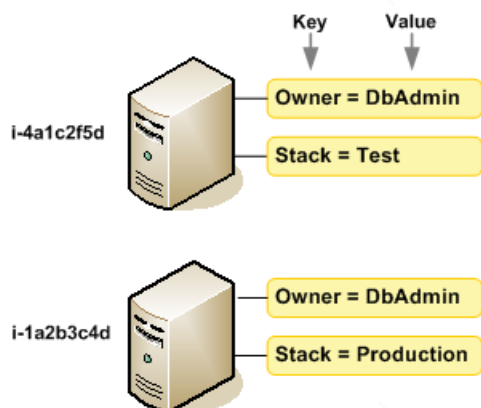
- [Tag Basics \(p. 428\)](#)

- [Tag Restrictions](#) (p. 428)
- [Tagging Your Resources for Billing](#) (p. 430)
- [Working with Tags in the Console](#) (p. 430)
- [API and CLI Overview](#) (p. 435)

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. 272).

Tag Restrictions

The following basic restrictions apply to tags:

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Tag Restrictions**

- Maximum number of tags per resource—10
- Maximum key length—127 Unicode characters
- Maximum value length—255 Unicode characters
- 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.

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 first get a list of those snapshots using `DescribeSnapshots` with a filter that specifies the tag. Then you use `DeleteSnapshots` with the resource identifiers of the snapshots (for example, `snap-1a2b3c4d`). 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 \(p. 425\)](#).

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	
Load Balancer	Yes	None
Network ACL	Yes	None
Network Interface	Yes	None
Placement Group	No	
Reserved Instance	Yes	None
Reserved Instance Listing	No	
Route Table	Yes	None
Spot Instance Request	Yes	None

Resource	Tagging support	Tagging restrictions
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
VPN Connection	Yes	None

For more information about tagging using the AWS console, see [Working with Tags in the Console \(p. 430\)](#). For more information about tagging using the API or command line, see [API and CLI Overview \(p. 435\)](#).

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. Then, 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*.

Working with Tags in 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.

Contents

- [Displaying Tags \(p. 430\)](#)
- [Adding and Deleting Tags on an Individual Resource \(p. 431\)](#)
- [Adding and Deleting Tags to a Group of Resources \(p. 432\)](#)
- [Adding a Tag When You Launch an Instance \(p. 434\)](#)
- [Filtering a List of Resources by Tag \(p. 434\)](#)

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

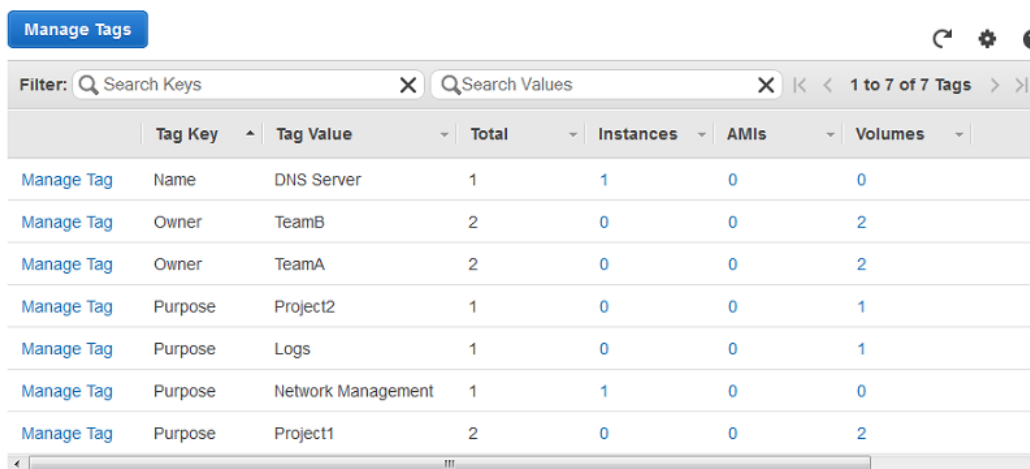
When you select a resource-specific page in the Amazon EC2 console, it displays a list of those resources. For example, if you select **Instances** from the navigation pane, the console displays a list of Amazon EC2 instances. When you select a resource from one of these lists (e.g., an instance), if 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.



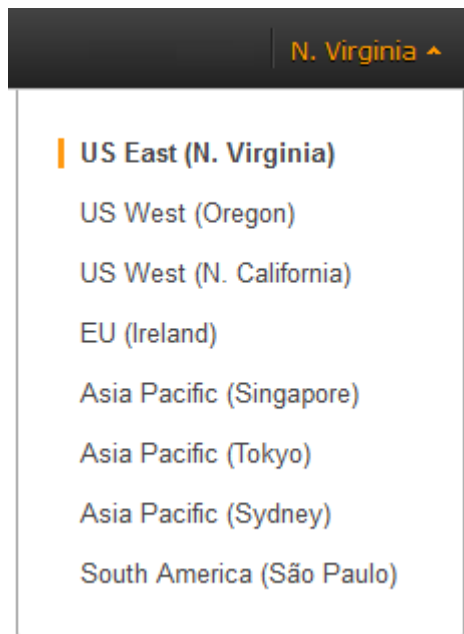
	Tag Key	Tag Value	Total	Instances	AMIs	Volumes
Manage Tag	Name	DNS Server	1	1	0	0
Manage Tag	Owner	TeamB	2	0	0	2
Manage Tag	Owner	TeamA	2	0	0	2
Manage Tag	Purpose	Project2	1	0	0	1
Manage Tag	Purpose	Logs	1	0	0	1
Manage Tag	Purpose	Network Management	1	1	0	0
Manage Tag	Purpose	Project1	2	0	0	2

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. 424).



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**.

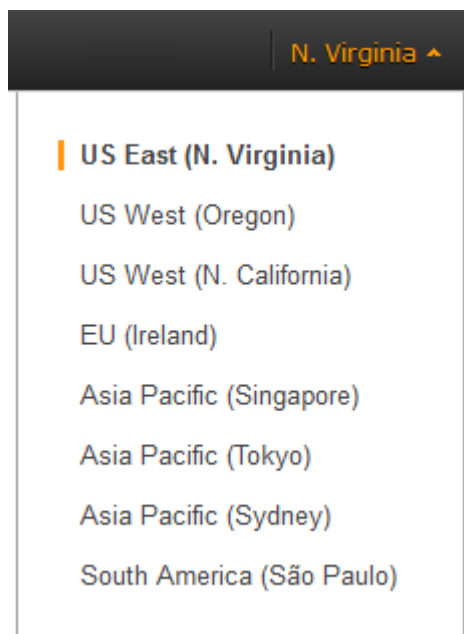
To delete a tag from an individual resource

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 \(p. 424\)](#).
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 \(p. 424\)](#).



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.

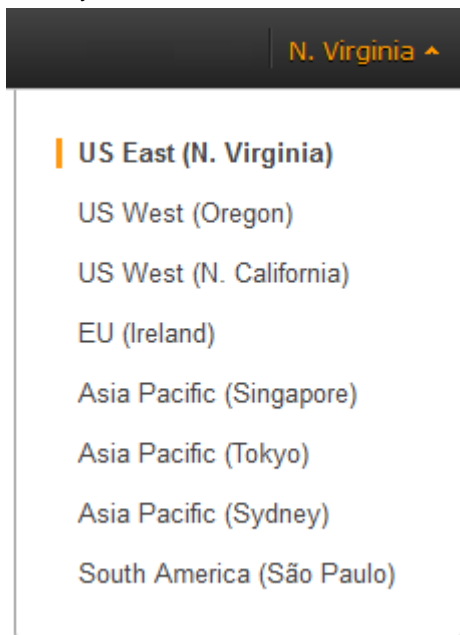
To remove a tag from a group of resources

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 \(p. 424\)](#).
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 \(p. 424\)](#).



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 an AMI \(p. 51\)](#).
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**.

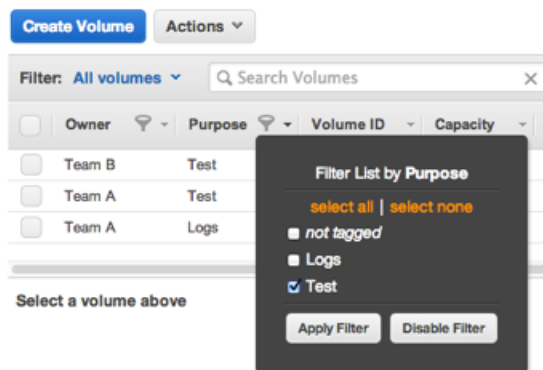
Filtering a List of Resources by Tag

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.



API and CLI Overview

Use the following API and CLI commands to add, update, list, and delete the tags for your resources. The documentation for each command provides examples. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

Description	Amazon EC2 CLI	AWS CLI	AWS Tools for Windows PowerShell	API Action
Adds or overwrites one or more tags for the specified resource or resources.	ec2-create-tags	create-tags	New-EC2Tag	CreateTags
Deletes the specified tags from the specified resource or resources.	ec2-delete-tags	delete-tags	Remove-EC2Tag	DeleteTags
Describes one or more tags for your resources.	ec2-describe-tags	describe-tags	Get-EC2Tag	DescribeTags

You can also filter a list of resources according to their tags. For example syntax, see [Filtering Resources Using the Console \(p. 426\)](#). 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 Elastic Compute Cloud 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 Elastic Compute Cloud API Reference](#).

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.

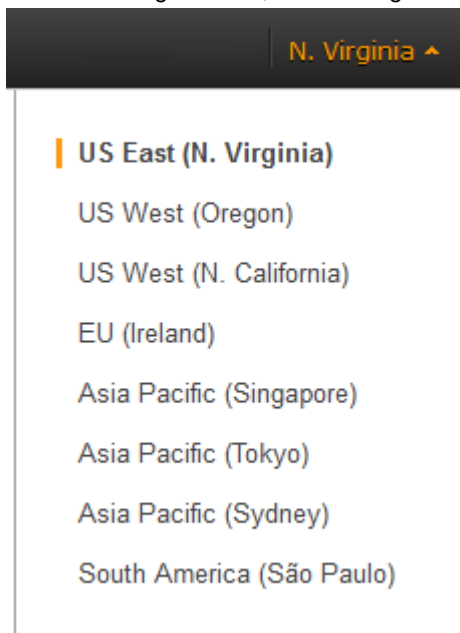
For more information about the limits for other services, see [AWS Service Limits](#) in the *Amazon Web Services General Reference*.

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. 437)
- [Getting Set Up for Usage Reports](#) (p. 437)
- [Granting IAM Users Access to the Amazon EC2 Usage Reports](#) (p. 439)
- [Instance Usage Report](#) (p. 439)
- [Reserved Instance Utilization Reports](#) (p. 443)

Available Reports

You can generate the following reports:

- [Instance usage report](#) (p. 439). This report covers your usage of on-demand instances, Spot Instances, and Reserved Instances.
- [Reserved Instances utilization report](#) (p. 443). 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. 427).
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 *Using IAM* 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?

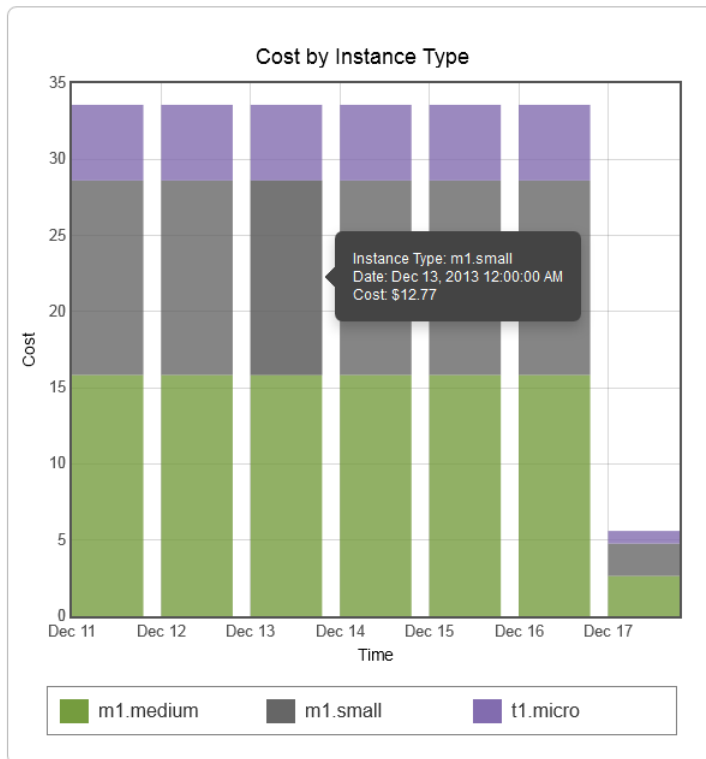
Topics

- [Report Formats \(p. 440\)](#)
- [Viewing Your Instance Usage \(p. 441\)](#)
- [Bookmarking a Customized Report \(p. 442\)](#)
- [Exporting Your Usage Data \(p. 442\)](#)

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 \(p. 437\)](#).

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**.

[EC2 Management Console](#) > [Reports](#) > EC2 Instance Usage

[Download](#) ▼

Granularity: [Daily](#) ▼ Time range: [Last 7 Days](#) ▼ Unit: [Cost](#) ▼

Filter: [Select](#) ▼ Group by: [Instance Type](#) ▼

Applied Filters: [Instance Type: t1.micro; m1.small; m1.medium;](#) [Clear All Filters](#) [Update Report](#)

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 \(p. 438\)](#).

Instance Usage Reports

Granularity: `Daily` Time range: `Last 14 Days` Unit: `Instance Hours`

Filter: `Select` Group by: `Tag` Key Name: `Project`

Applied Filters: None

[Clear All Filters](#) [Update Report](#)

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 single day 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 \(p. 437\)](#)). 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.

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 \(p. 437\)](#).

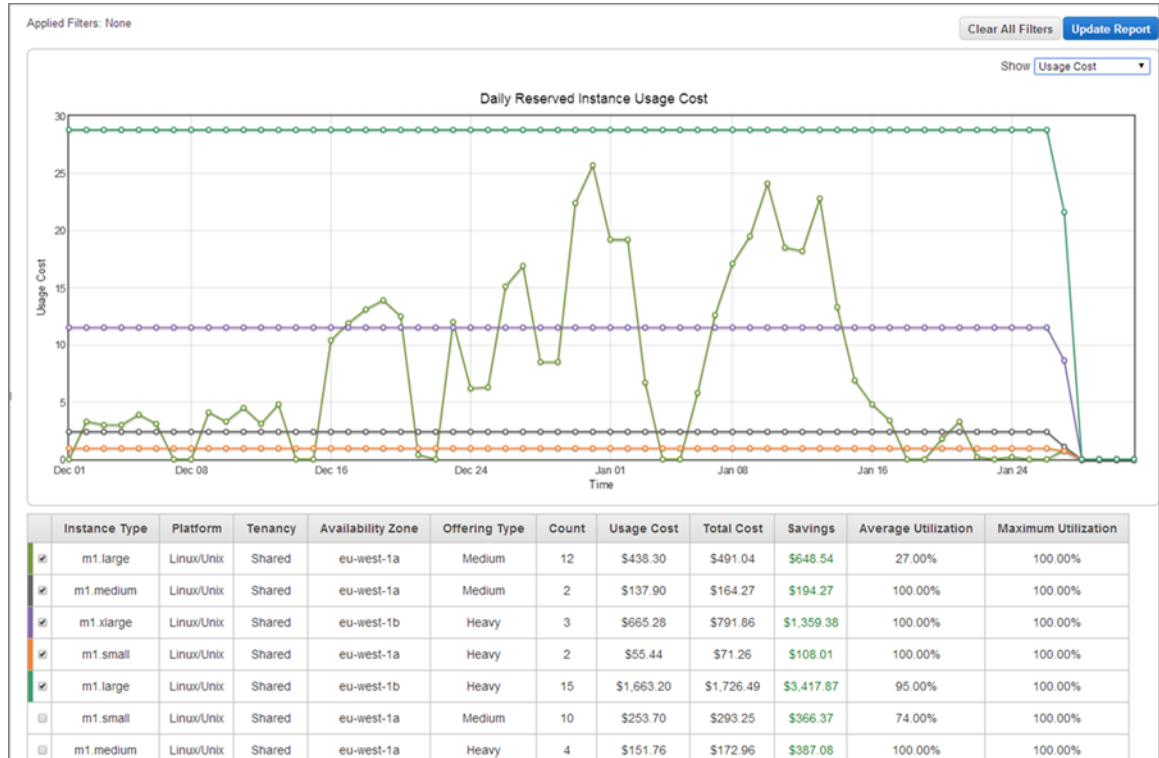
Topics

- [Getting to Know the Report \(p. 443\)](#)
- [Bookmarking a Customized Report \(p. 447\)](#)
- [Exporting Your Usage Data \(p. 447\)](#)
- [Options Reference \(p. 447\)](#)

Getting to Know the Report

The Reserved Instance utilization report displays your requested utilization data in graph and table formats.

Amazon Elastic Compute Cloud User Guide for Microsoft Windows Reserved Instance Utilization



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 \(p. 447\)](#).

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.

Instance Type	Platform	Tenancy	Availability Zone	Offering Type	Count	Usage Cost	Total Cost	Savings	Average Utilization	Maximum Utilization
<input checked="" type="checkbox"/> m1.large	Linux/Unix	Shared	eu-west-1a	Medium	12	\$438.30	\$491.04	\$648.54	27.00%	100.00%

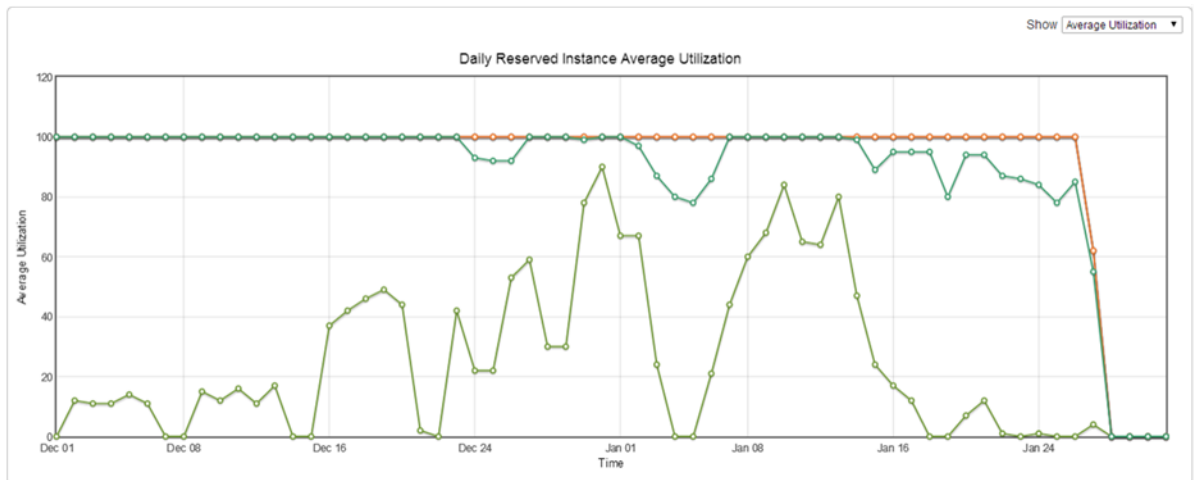
Amazon Elastic Compute Cloud User Guide for Microsoft Windows Reserved Instance Utilization

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 a day to weeks, months, or years.

For example, the following report table shows Reserved Instance utilization for a two-month period. In this case, the period is December 1 through January 31. The report has been filtered to only return data about Reserved Instance buckets in eu-west-1a, eu-west-1b, and eu-west-1c.

	Instance Type	Platform	Tenancy	Availability Zone	Offering Type	Count	Usage Cost	Total Cost	Savings	Average Utilization	Maximum Utilization
<input checked="" type="checkbox"/>	m1.large	Linux/Unix	Shared	eu-west-1a	Medium	12	\$438.30	\$491.04	\$648.54	27.00%	100.00%
<input checked="" type="checkbox"/>	m1.medium	Linux/Unix	Shared	eu-west-1a	Medium	2	\$137.90	\$164.27	\$194.27	100.00%	100.00%
<input checked="" type="checkbox"/>	m1.xlarge	Linux/Unix	Shared	eu-west-1b	Heavy	3	\$665.28	\$791.86	\$1,359.38	100.00%	100.00%
<input checked="" type="checkbox"/>	m1.small	Linux/Unix	Shared	eu-west-1a	Heavy	2	\$55.44	\$71.26	\$108.01	100.00%	100.00%
<input checked="" type="checkbox"/>	m1.large	Linux/Unix	Shared	eu-west-1b	Heavy	15	\$1,663.20	\$1,726.49	\$3,417.87	95.00%	100.00%
<input type="checkbox"/>	m1.small	Linux/Unix	Shared	eu-west-1a	Medium	10	\$253.70	\$293.25	\$366.37	74.00%	100.00%
<input type="checkbox"/>	m1.medium	Linux/Unix	Shared	eu-west-1a	Heavy	4	\$151.76	\$172.96	\$387.08	100.00%	100.00%
<input type="checkbox"/>	m1.large	Linux/Unix	Shared	eu-west-1a	Heavy	17	\$1,884.96	\$1,948.25	\$3,663.33	92.00%	100.00%

The graph shows the daily Reserved Instance average utilization for the selected buckets.



Customizing the Report

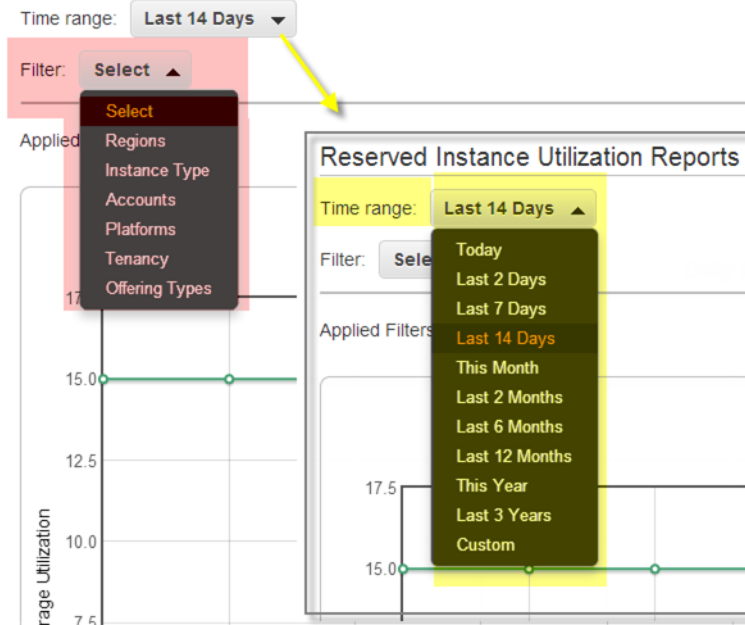
You can customize the Reserved Instance utilization report with **Time range** and **Filter** options.

Amazon Elastic Compute Cloud User Guide for Microsoft Windows

Reserved Instance Utilization

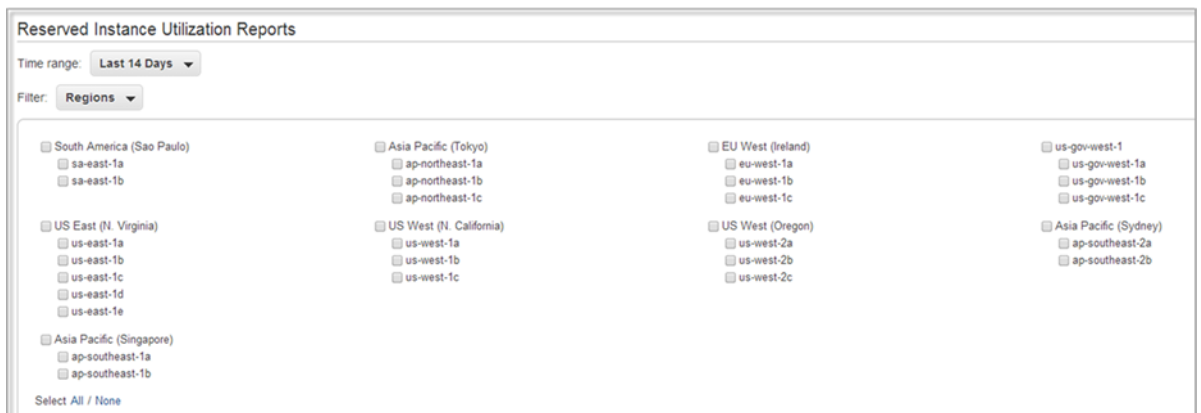
[Billing & Cost Management](#) > [Reports](#) > [EC2 Reserved Instance Utilization](#)

Reserved Instance Utilization Reports



Time range provides a list of common relative time ranges, ranging from **Today** 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, account, platform, tenancy, and offering type. 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. 447).

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.

- Usage Cost

Shows the total cost based on hourly fees for a selected bucket of Reserved Instances. For heavy utilization Reserved Instance buckets, usage cost is calculated by multiplying the number of hours in the time frame by the hourly rates associated with the Reserved Instances in the bucket. For medium and light utilization Reserved instances, usage cost is calculated by multiplying the number of instance hours used for the selected time frame by the hourly rates for the respective Reserved Instances in the bucket.

- 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.

Use **Time range** to specify the period on which the report will be based.

Note

All times are specified in UTC time.

- Today

Shows data for usage that takes place during any of the hours for the current calendar day. Can be used with hourly, daily, or monthly granularities.

- Last 2 Days

Shows data for usage that took place during the current and previous one calendar day. Can be used with hourly, daily, or monthly granularities.

- 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 2 Months

Shows data for usage that took place during the current and previous 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 Types
- Accounts
- Platforms
- Tenancy
- Offering Type

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.

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 AWS resources:

- EC2 instances
- EBS volumes
- ELB load balancers
- Auto Scaling groups and Availability Zones
- AWS Elastic Beanstalk applications
- AWS CloudFormation stacks

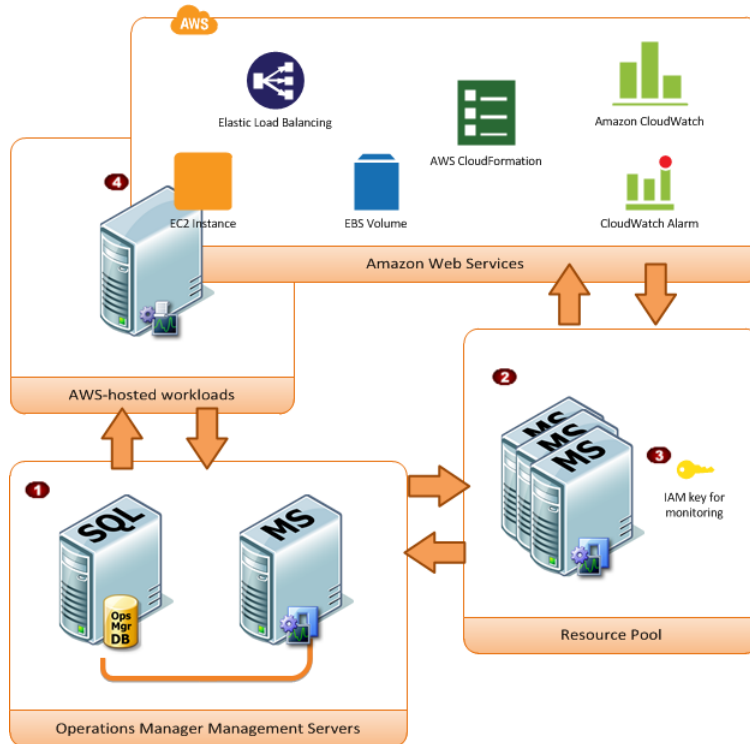
Contents

- [Overview of AWS Management Pack for System Center 2012 \(p. 450\)](#)
- [Overview of AWS Management Pack for System Center 2007 R2 \(p. 451\)](#)
- [Downloading the AWS Management Pack \(p. 452\)](#)
- [Deploying the AWS Management Pack \(p. 453\)](#)
- [Using the AWS Management Pack \(p. 462\)](#)
- [Troubleshooting the AWS Management Pack \(p. 480\)](#)

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 for System Center 2012.



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	Resource pool	One or more management servers used for communicating with AWS using the AWS SDK for .NET. These servers must have Internet connectivity.
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 <i>Using IAM</i> .

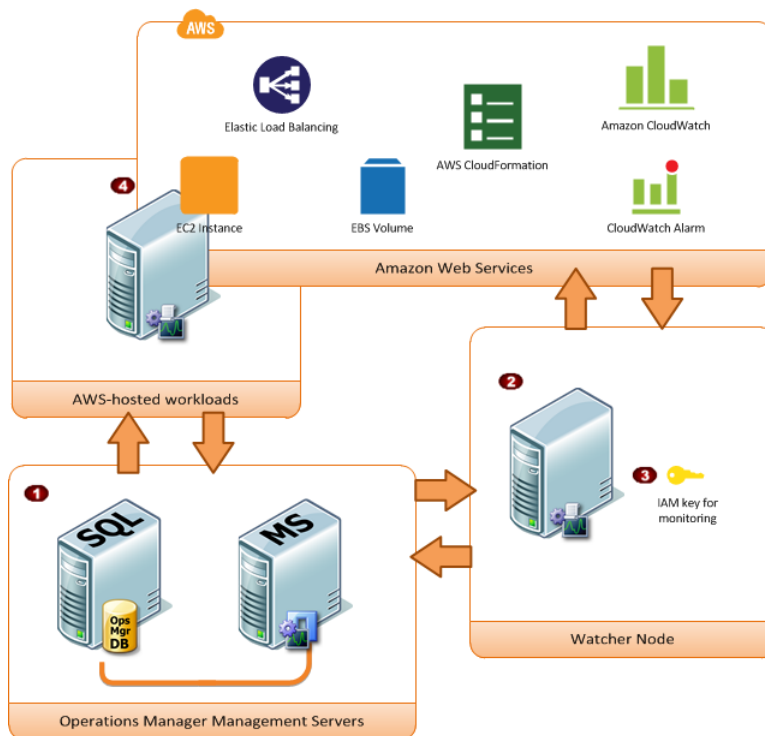
**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Overview of AWS Management Pack for System Center
2007 R2**

Item	Component	Description
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 uses a designated computer in your data center that has Internet access, called a *watcher node*, and AWS APIs to remotely discover and collect information about your AWS resources.

The following diagram shows the main components of AWS Management Pack for System Center 2007 R2.



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.

Item	Component	Description
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.
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 <i>Using IAM</i> .
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 Requirements

Before you download the AWS Management Pack, ensure that your systems meet the following requirements:

- System Center Operations Manager 2012 R2, System Center Operations Manager 2012 SP1, or System Center Operations Manager 2007 R2
- [System Center 2012] 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:
 - [System Center 2012] version 7.3.2026.0 or later
 - [System Center 2007] version 6.1.7000.256 or later

Prerequisites

Before you download the AWS Management Pack, ensure that your systems meet the following prerequisites:

- [System Center 2012] 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.

- [System Center 2007 R2] 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.
- [System Center 2012] The action account for the management server must have local administrator privileges on the management server.
- [System Center 2007 R2] The action account for the watcher node must have local administrator privileges on the watcher node.
- The Amazon CloudWatch service must be enabled for your AWS account.
- The instances to be monitored must run System Center Operations Manager agents. If you use this feature, you must ensure that the agents are deployed, running, and can communicate with the management servers in your data center.

To download the AWS Management Pack

1. On the [AWS Management Pack for Microsoft System Center](#) website, click either **SCOM 2012 / SCOM 2012 R2 MP** or **SCOM 2007 R2 MP**.
2. [System Center 2012] Download `AWS-SCOM-MP-2.0.zip` to your computer and unzip it.
3. [System Center 2007 R2] Save `AWS_MP_Setup.msi` to your computer.

The next step is to import `Amazon.AmazonWebServices.mpb`. For more information, see [Deploying the AWS Management Pack](#) (p. 453).

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. 452).

Tasks

- [Step 1: Installing the AWS Management Pack](#) (p. 453)
- [Step 2: Configuring the Watcher Node](#) (p. 455)
- [Step 3: Create an AWS Run As Account](#) (p. 455)
- [Step 4: Run the Add Monitoring Wizard](#) (p. 458)

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 for System Center 2012

1. In the System Center Operations Manager Operations Console, on the **Go** menu, click **Administration**.
2. Right-click **Management Packs**, and then click **Import Management Packs**.
3. In the **Import Management Packs Wizard**, click **Add**, and then click **Add from disk**.
4. In the **Select Management Packs to import** dialog box, click **Amazon.AmazonWebServices.mpb** to import from the directory you downloaded it in, and then click **Open**.

5. On the **Select Management Packs** page, the AWS Management Pack that you selected for import is listed. Click **Import**.

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 for System Center 2007 R2

The management pack is distributed as a Microsoft System Installer file, `AWS_MP_Setup.msi`. It contains the required DLLs for the watcher node and System Center Operations Manager Root 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 AWS 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` and registers it in the GAC.

AWS Watcher Node

Installs `Amazon.AmazonWebServices.Modules.dll` and registers it in the GAC, and then installs the AWS SDK for .NET (`AWSSDK.dll`) into 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. You must manually import the management pack into System Center, just like any other management pack.

7. In the Operations Console, on the **Go** menu, click **Administration**.
8. In the **Administration** navigation pane, right-click **Administration**, and then click **Import Management Packs**.
9. In the **Import Management Packs** wizard, 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`, the directory that contains your management pack file, click `Amazon.AmazonWebServices.mp`, and then click **Open**.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Step 2: Configuring the Watcher Node**

11. On the **Select Management Packs** page, in the **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**.

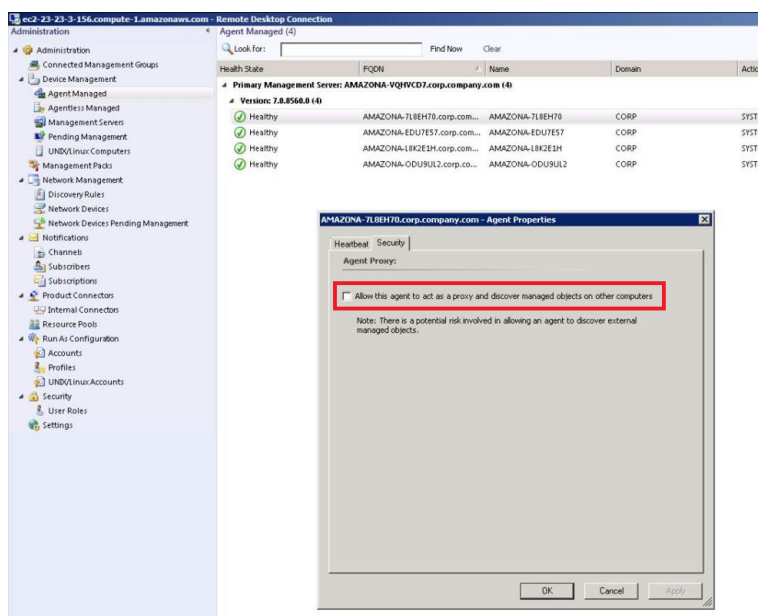
Step 2: Configuring the Watcher Node

On System Center Operations Manager 2007, 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.

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

1. In the System Center Operations Manager 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**.



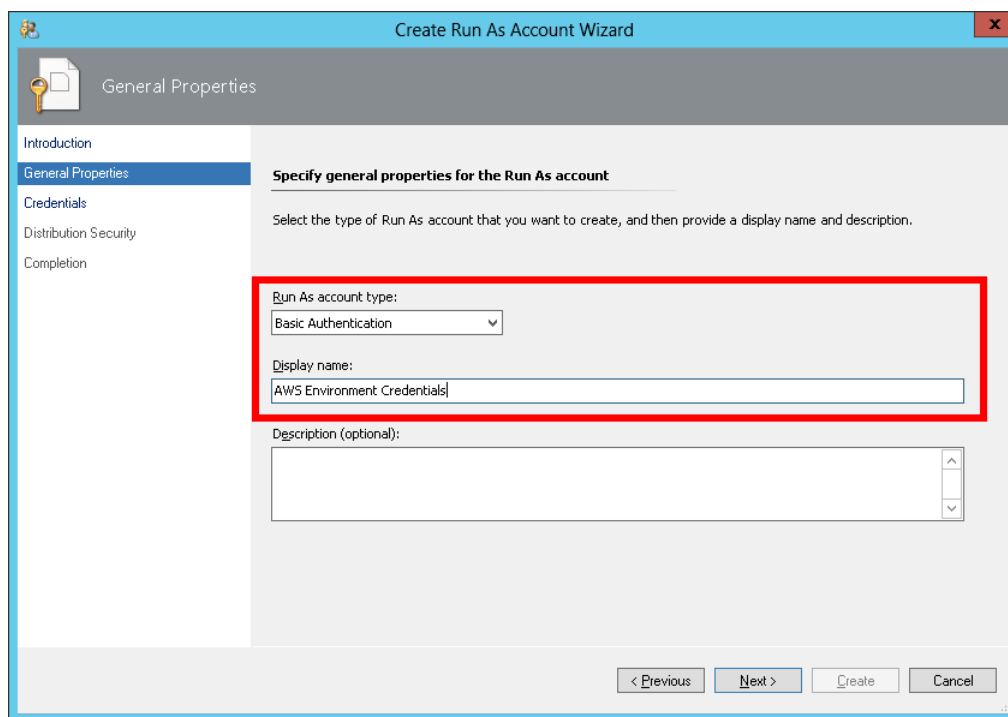
Step 3: Create an AWS Run As Account

You must set up credentials that grant AWS Management Pack access to your AWS resources.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Step 3: Create an AWS Run As Account**

To create an AWS Run As account

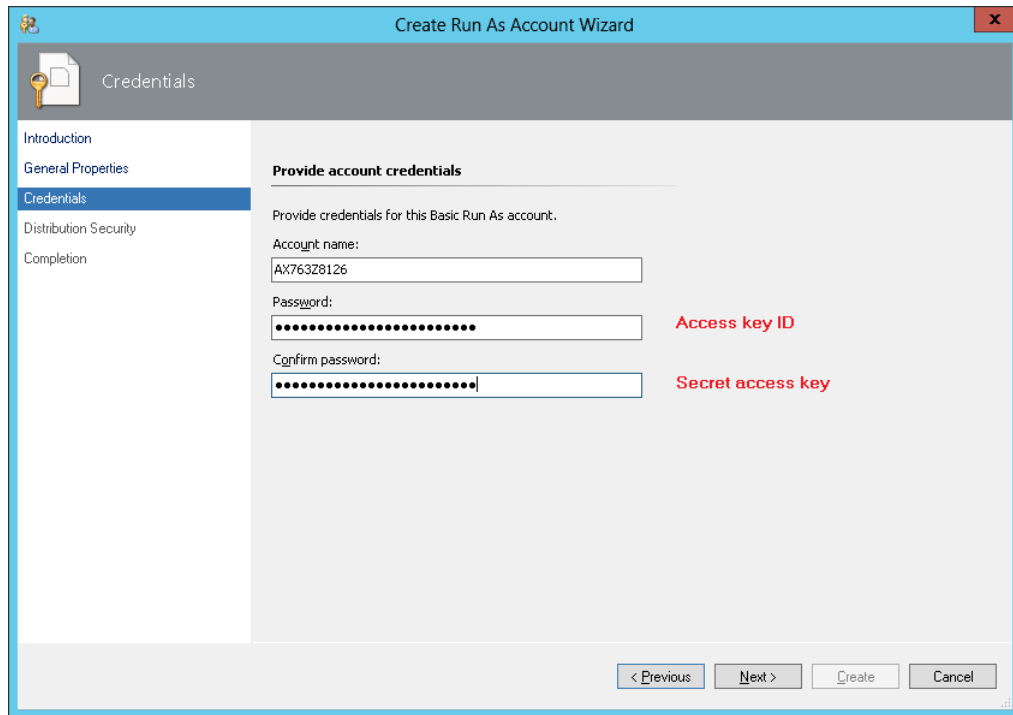
1. We recommend that you create an IAM user with the minimum access rights required (for example, the **Read Only Access** policy template works in most cases). You'll need the access keys (access key ID and secret access key) for this user to complete this procedure. For more information, see [Administering Access Keys for IAM Users](#) in Using IAM.
2. In the System Center Operations Manager 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**.



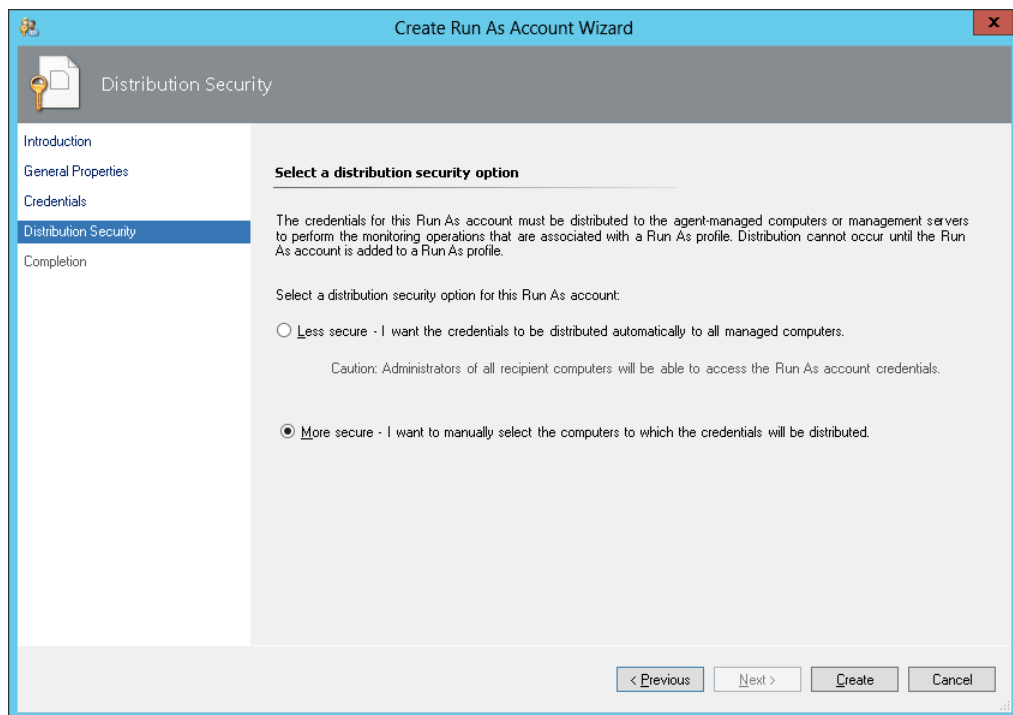
The screenshot shows the 'Create Run As Account Wizard' dialog box with the 'General Properties' page selected. The 'Run As account type' dropdown menu is set to 'Basic Authentication'. The 'Display name' text box contains the text 'AWS Environment Credentials'. A red rectangular box highlights the 'Run As account type' dropdown and the 'Display name' text box. The 'Description (optional)' text box is empty. At the bottom of the dialog, there are four buttons: '< Previous', 'Next >', 'Create', and 'Cancel'.

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**.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Step 3: Create an AWS Run As Account**



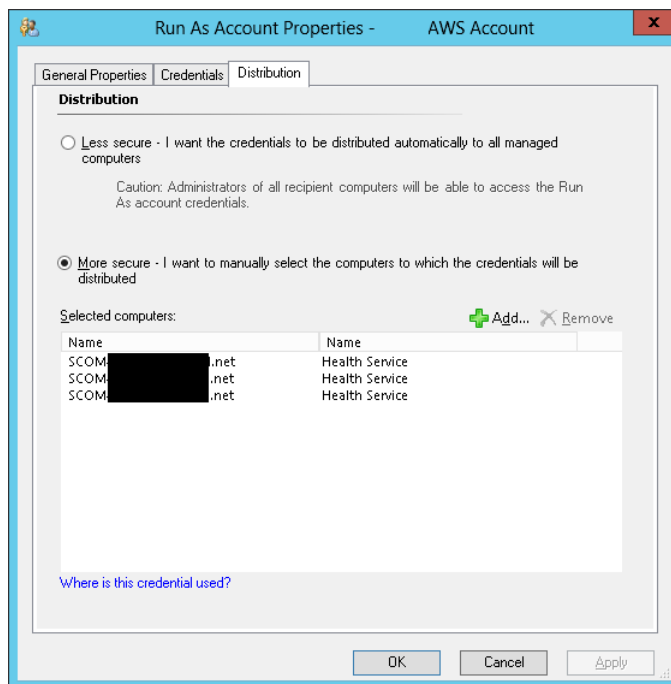
8. On the **Distribution Security** page, select **More secure - I want to manually select the computers to which the credentials will be distributed**, and then click **Create**.



9. Click **Close**.
10. In the list of accounts, select the account that you just created.
11. In the **Actions** pane, click **Properties**.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Step 4: Run the Add Monitoring Wizard**

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

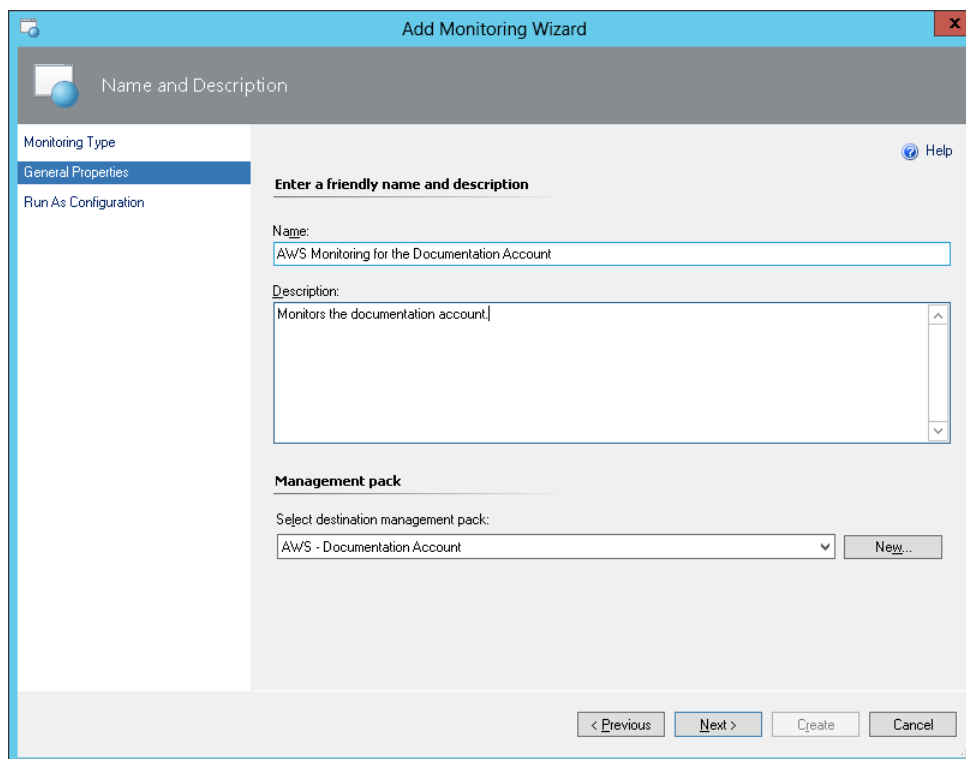
You can configure the AWS Management Pack to monitor a particular AWS account by using the Add Monitoring Wizard, which is available in the **Authoring** workspace of the Operations Console. This wizard creates a management pack that contains the settings for the AWS account to monitor. You must run 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 System Center Operations Manager 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**.

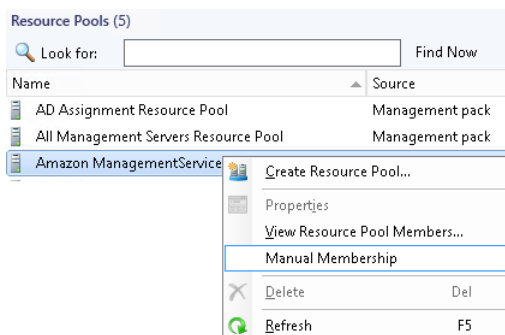
**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Step 4: Run the Add Monitoring Wizard**



Note

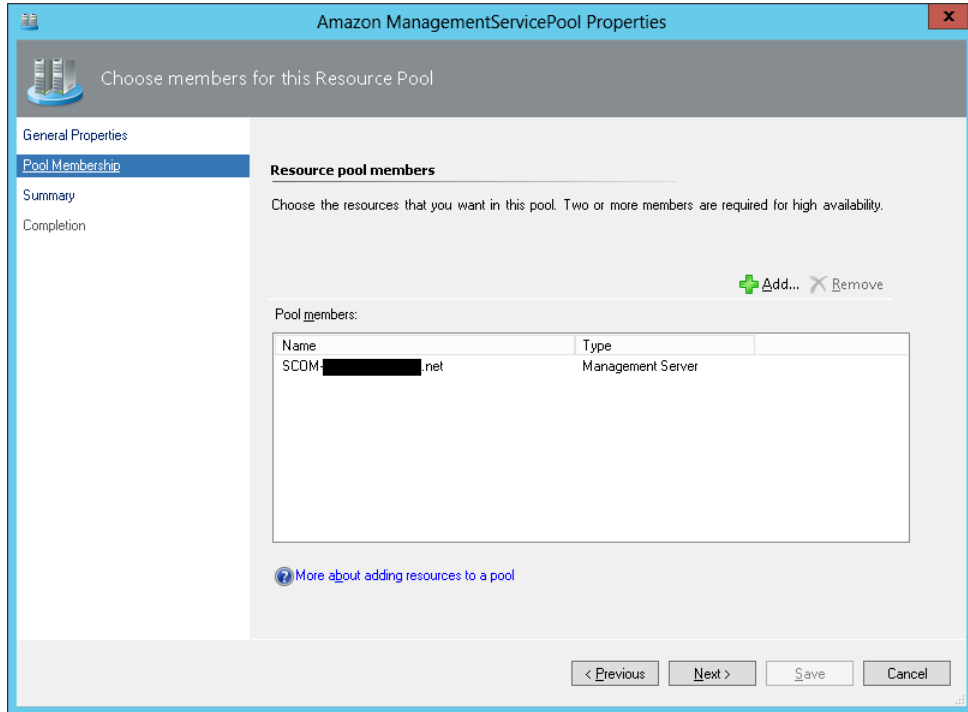
By default, when you create a management pack object, disable a rule or monitor, or create an override, Operations Manager saves the setting to the default management pack. As a best practice, you should create a separate management pack for each sealed management pack that you want to customize, instead 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**.

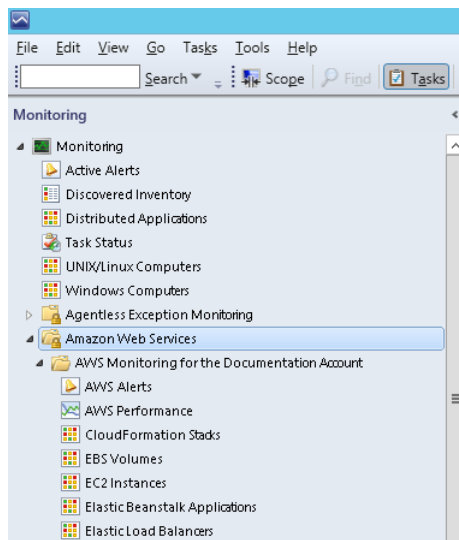


- d. Right-click the **AWS Resource Pool** in the **Resource Pools** pane and select **Properties**.
 - e. On the **Pool Membership** page, remove the management servers that should not monitor AWS resources.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Step 4: Run the Add Monitoring Wizard**



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

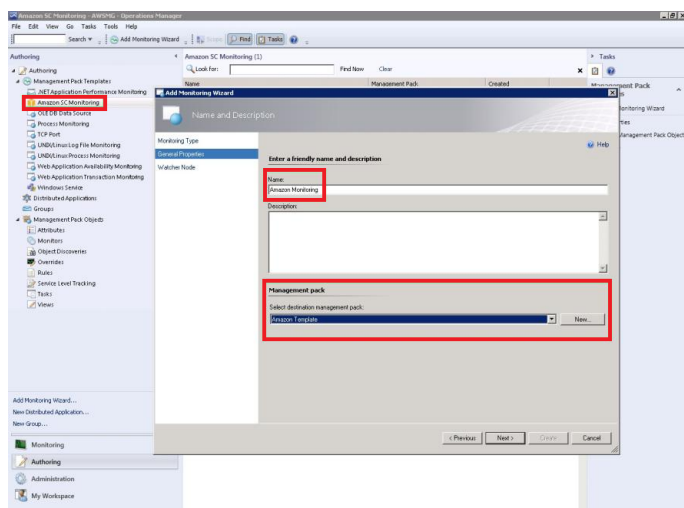
To run the Add Monitoring Wizard on System Center Operations Manager 2007

1. In the System Center Operations Manager **Operations Console**, on the **Go** menu, click **Authoring**.

Amazon Elastic Compute Cloud User Guide for Microsoft Windows

Step 4: Run the Add Monitoring Wizard

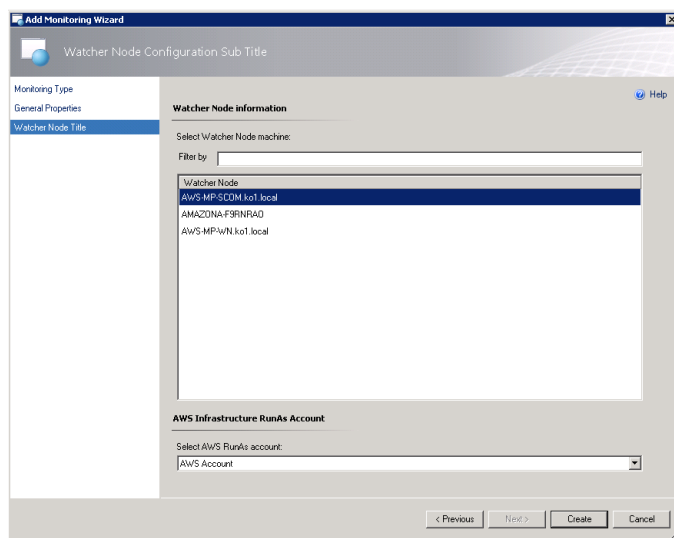
2. In the **Authoring** workspace, expand the **Management Pack Templates** node, right-click **Amazon Web Services**, and then click **Add Monitoring Wizard**.
3. In the **Add Monitoring Wizard**, in the **Select the monitoring type list**, select **Amazon Web Services**, and then click **Next**.
4. 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.

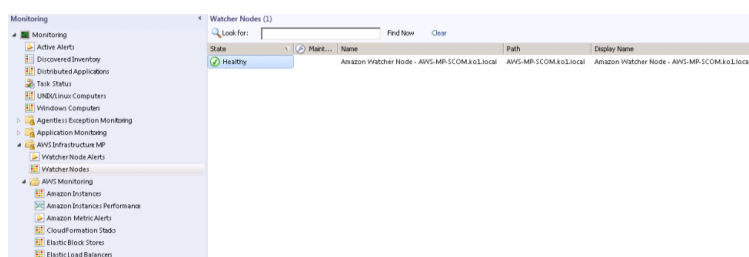
6. On the **Watcher Node Configuration** page, in the **Watcher Node** list, select an agent-managed computer to act as the watcher node.



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 \(p. 480\)](#).



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 \(p. 480\)](#).

Using the AWS Management Pack

You can use the AWS Management Pack to monitor the health of your AWS resources.

Contents

- [Views \(p. 462\)](#)
- [Discoveries \(p. 472\)](#)
- [Monitors \(p. 474\)](#)
- [Rules \(p. 475\)](#)
- [Events \(p. 478\)](#)
- [Health Model \(p. 479\)](#)
- [Customizing the AWS Management Pack \(p. 480\)](#)

Views

The AWS Management Pack provides the following views, which are displayed in the **Monitoring** workspace of the Operations Console.

Views

- [EC2 Instances](#) (p. 463)
- [Amazon Instances Performance](#) (p. 464)
- [EBS Volumes](#) (p. 465)
- [CloudWatch Alarms](#) (p. 466)
- [Elastic Load Balancers](#) (p. 467)
- [Elastic Beanstalk Applications](#) (p. 468)
- [CloudFormation Stacks](#) (p. 470)
- [Watcher Nodes \(System Center Operations Manager 2007\)](#) (p. 472)

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. The **Windows Computer** and **UNIX/Linux Computer** columns help you determine whether Operations Manager Agent is running inside the instance.

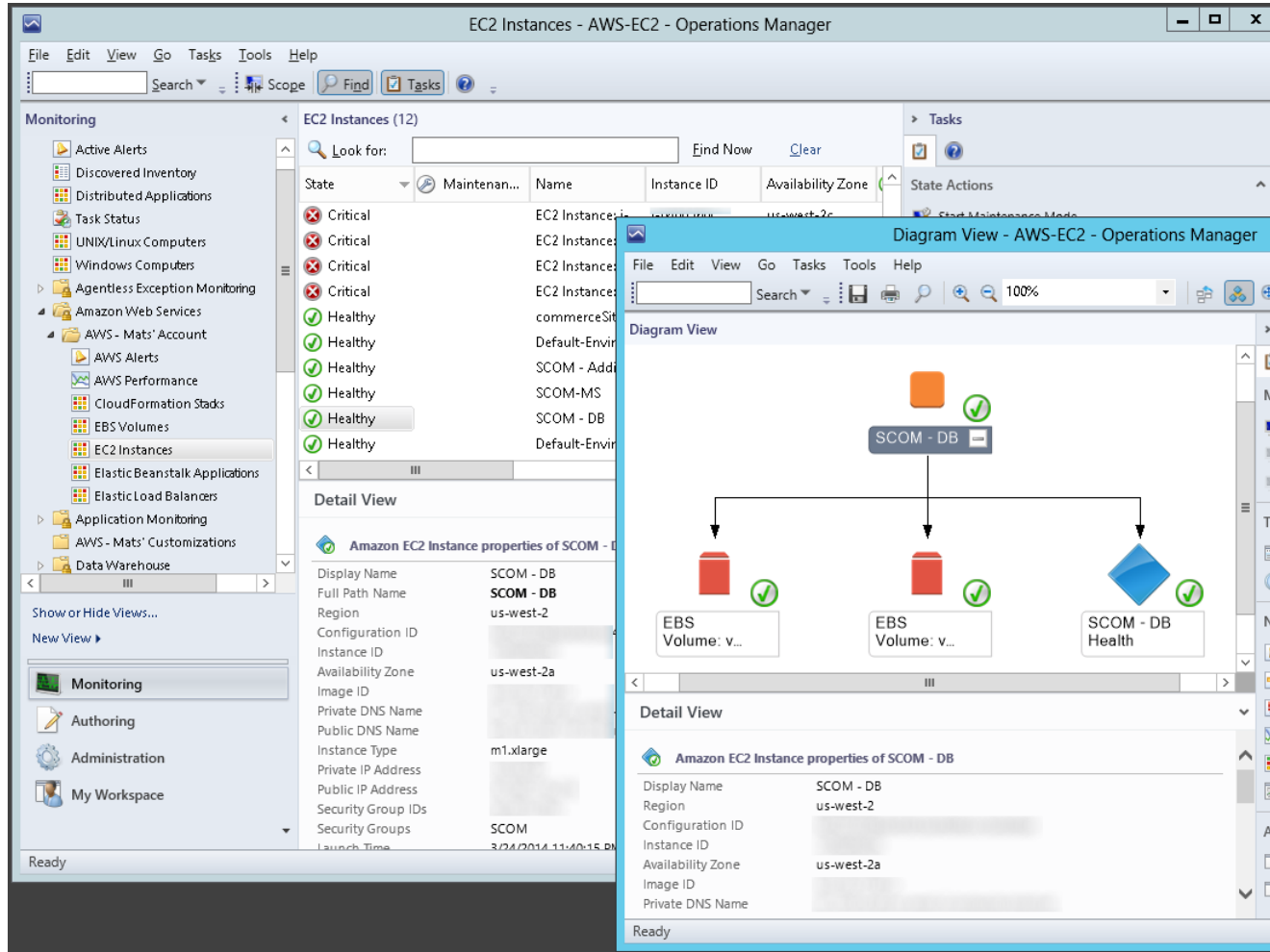
When you select an EC2 instance, you can perform instance health tasks.

- **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.

EC2 Instances Diagram View

Shows the relationship of an instance with other components.

Amazon Elastic Compute Cloud User Guide for Microsoft Windows Views

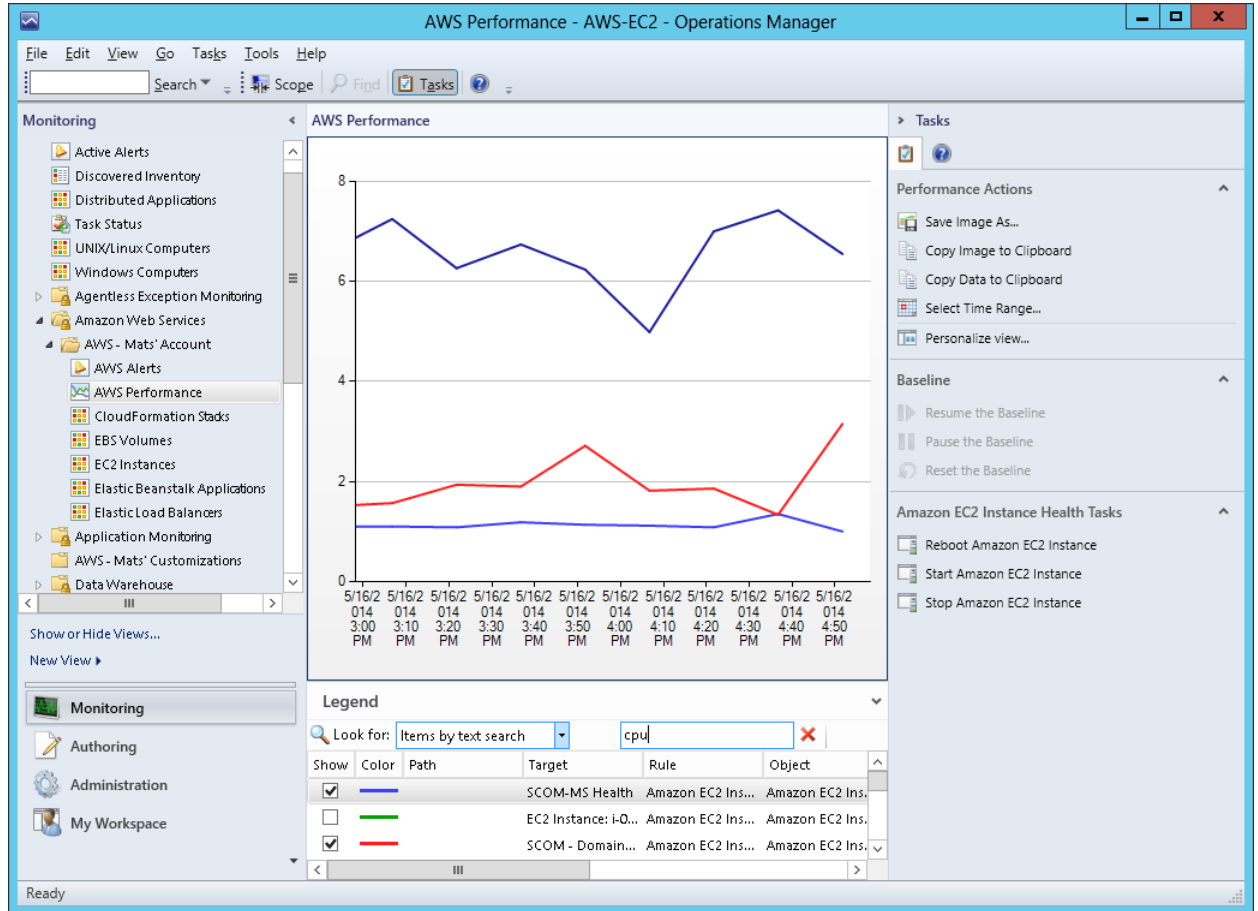


Amazon Instances Performance

Shows the default Amazon CloudWatch metrics for Amazon EC2, Amazon EBS, and Elastic Load Balancing. 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 Elastic Compute Cloud User Guide for Microsoft Windows Views



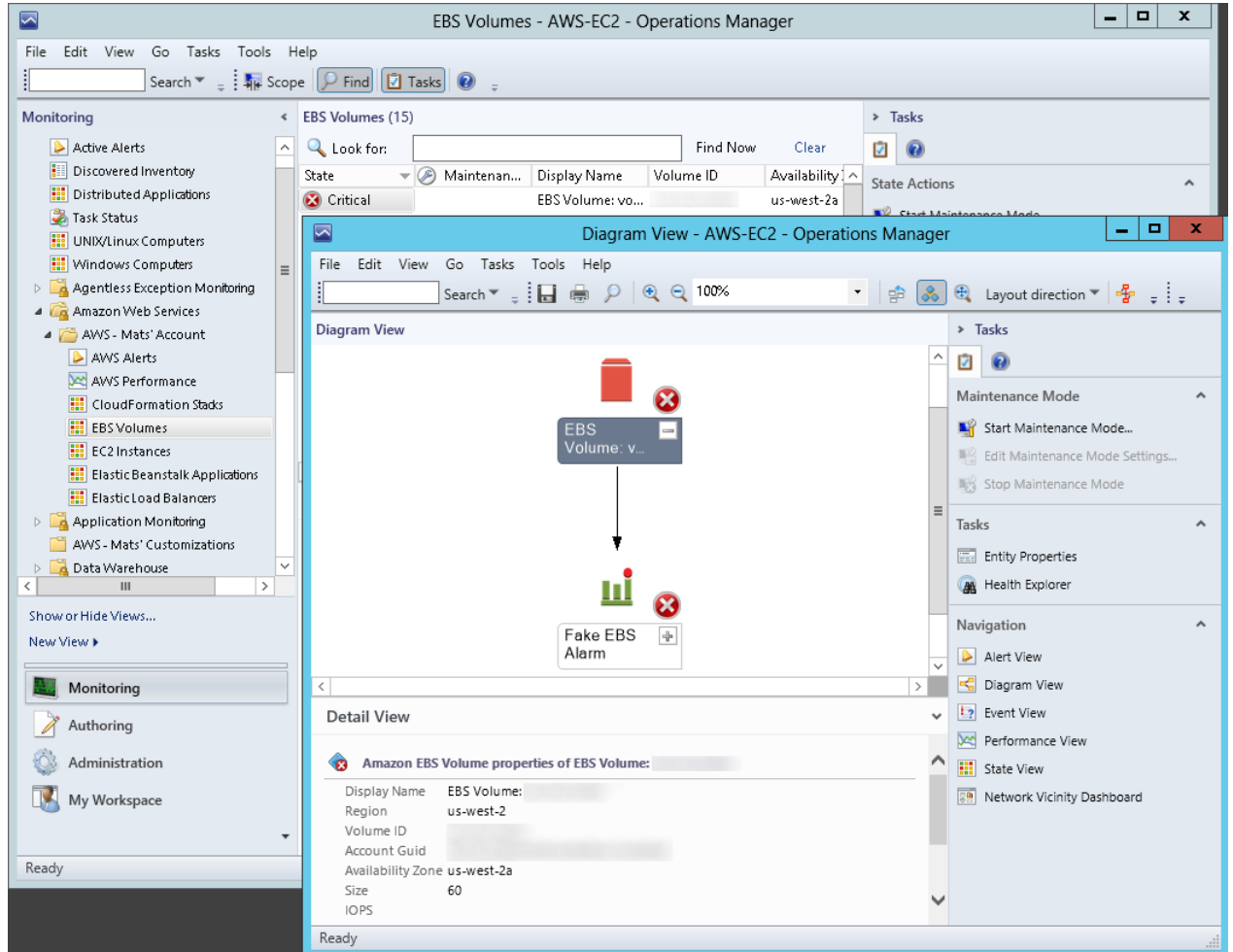
EBS Volumes

Shows the health state of all the Amazon EBS volumes for a particular AWS account from all Availability Zones and regions.

EBS Volumes Diagram View

Shows an Amazon EBS volume and any associated alarms. The following illustration shows an example:

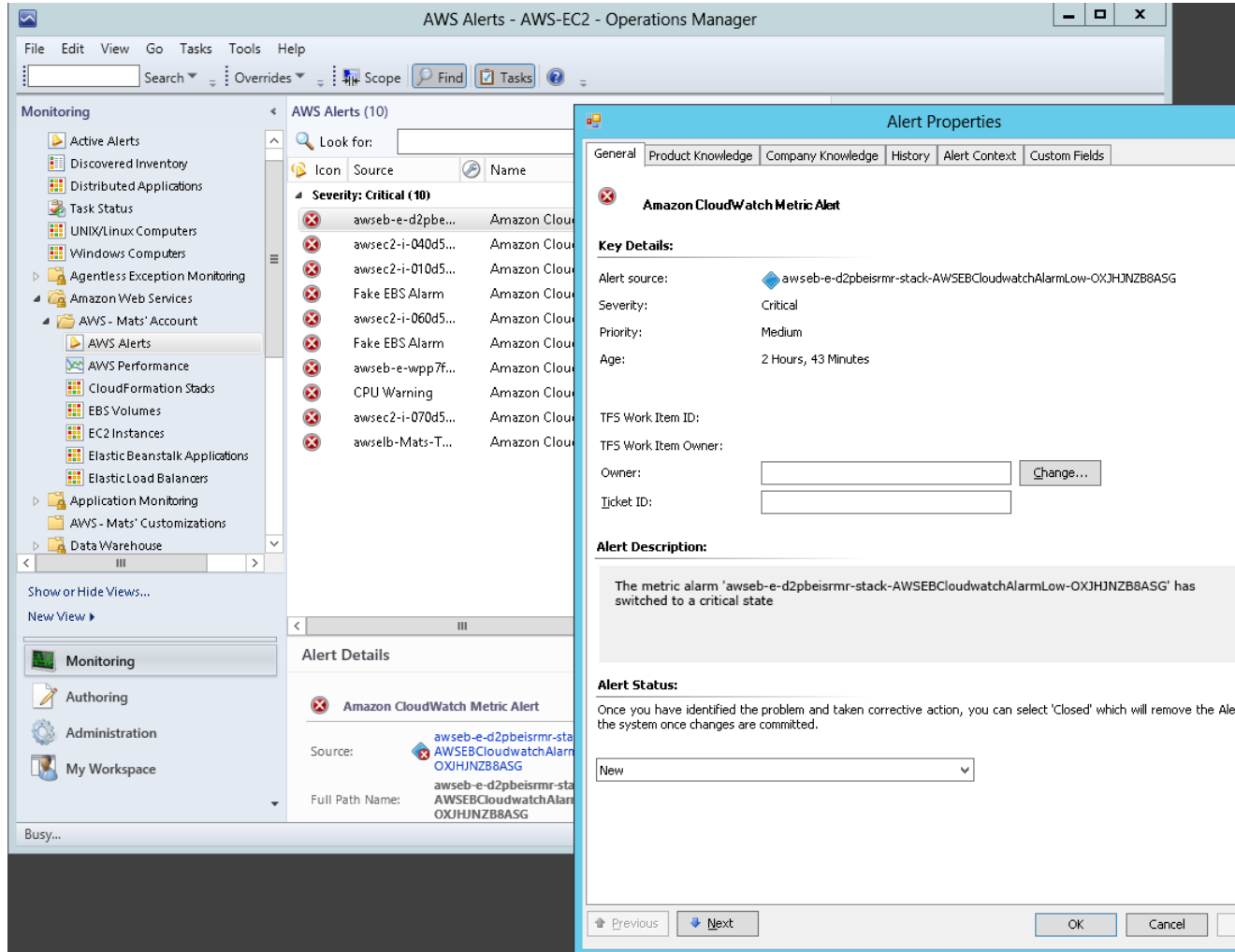
Amazon Elastic Compute Cloud User Guide for Microsoft Windows Views



CloudWatch Alarms

Shows Amazon CloudWatch alarms related to the discovered AWS resources.

Amazon Elastic Compute Cloud User Guide for Microsoft Windows Views



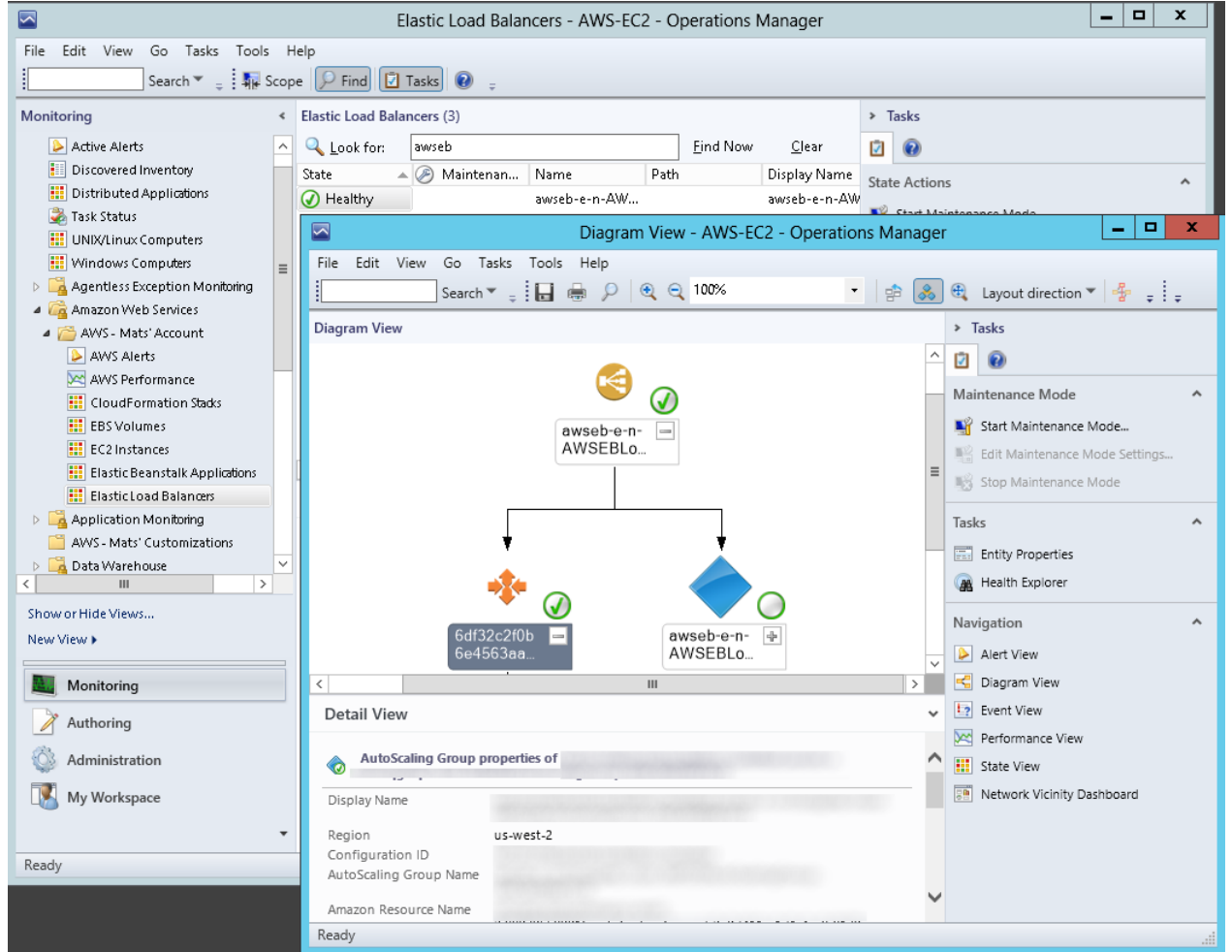
Elastic Load Balancers

Shows the health state of all the load balancers for a particular AWS account from all regions.

Elastic Load Balancer Diagram View

Shows the Elastic Load Balancing relationship with other components. The following illustration shows an example:

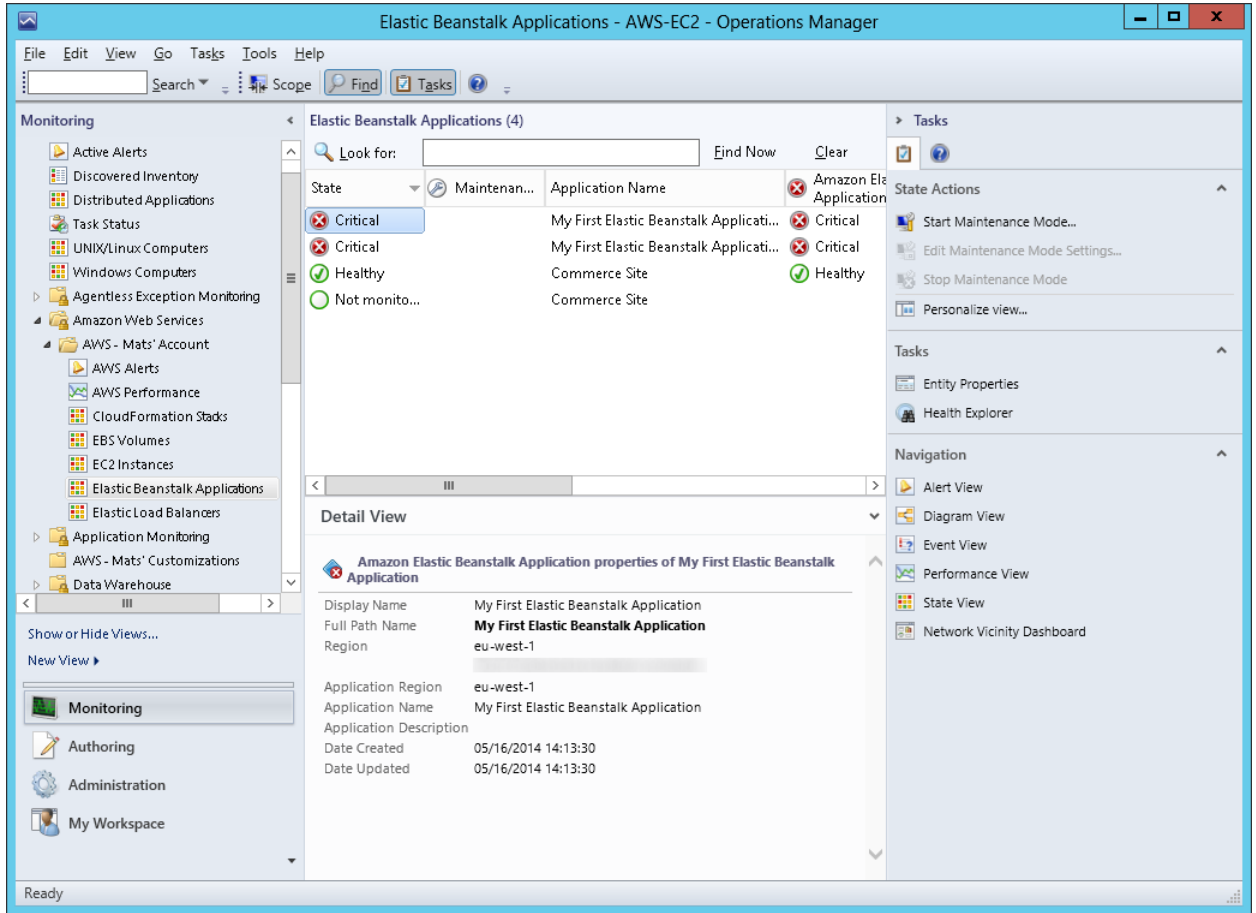
Amazon Elastic Compute Cloud User Guide for Microsoft Windows Views



Elastic Beanstalk Applications

Shows the state of all discovered AWS Elastic Beanstalk applications.

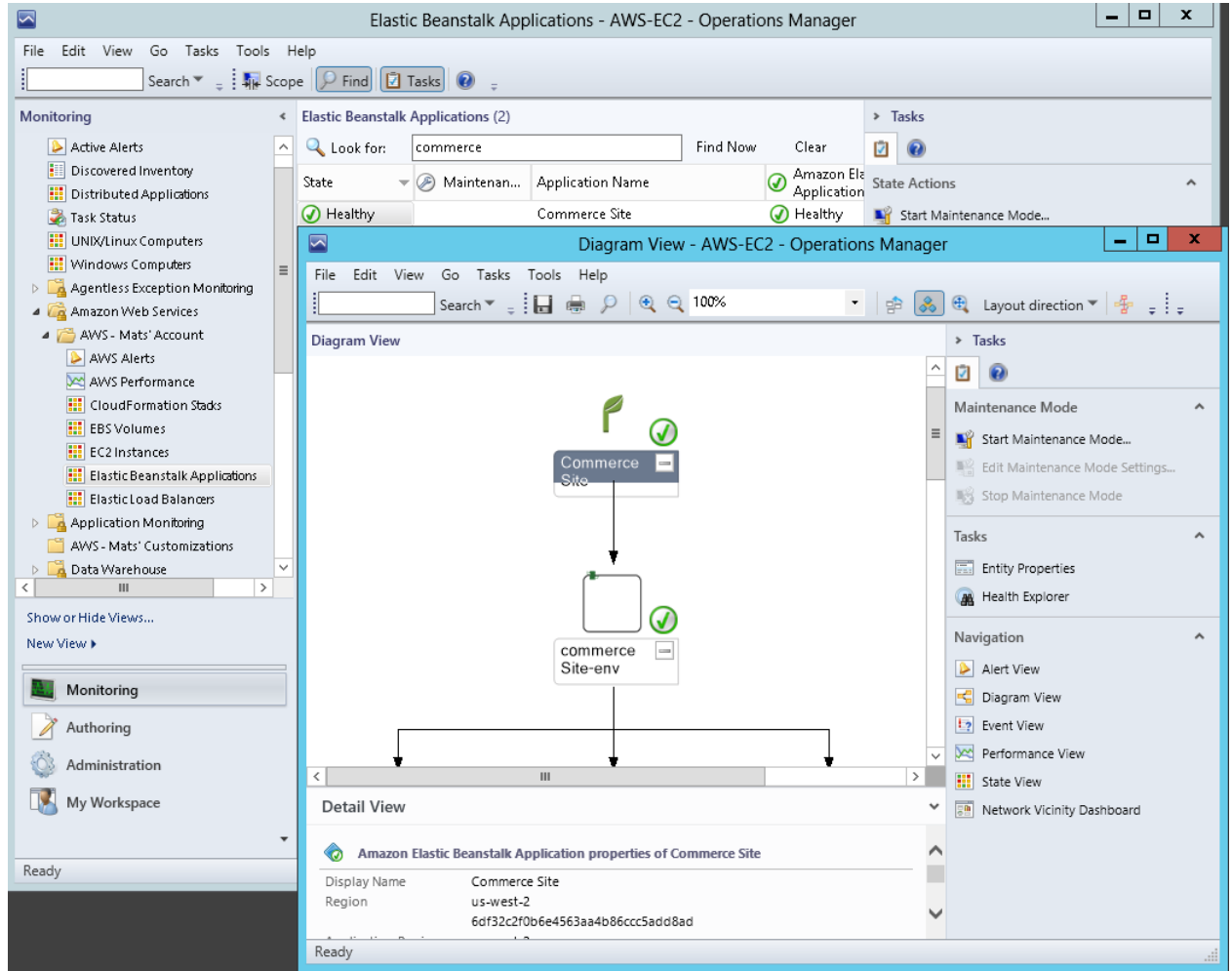
Amazon Elastic Compute Cloud User Guide for Microsoft Windows Views



Elastic Beanstalk Applications Diagram View

Shows the AWS Elastic Beanstalk application, application environment, application configuration, and application resources objects.

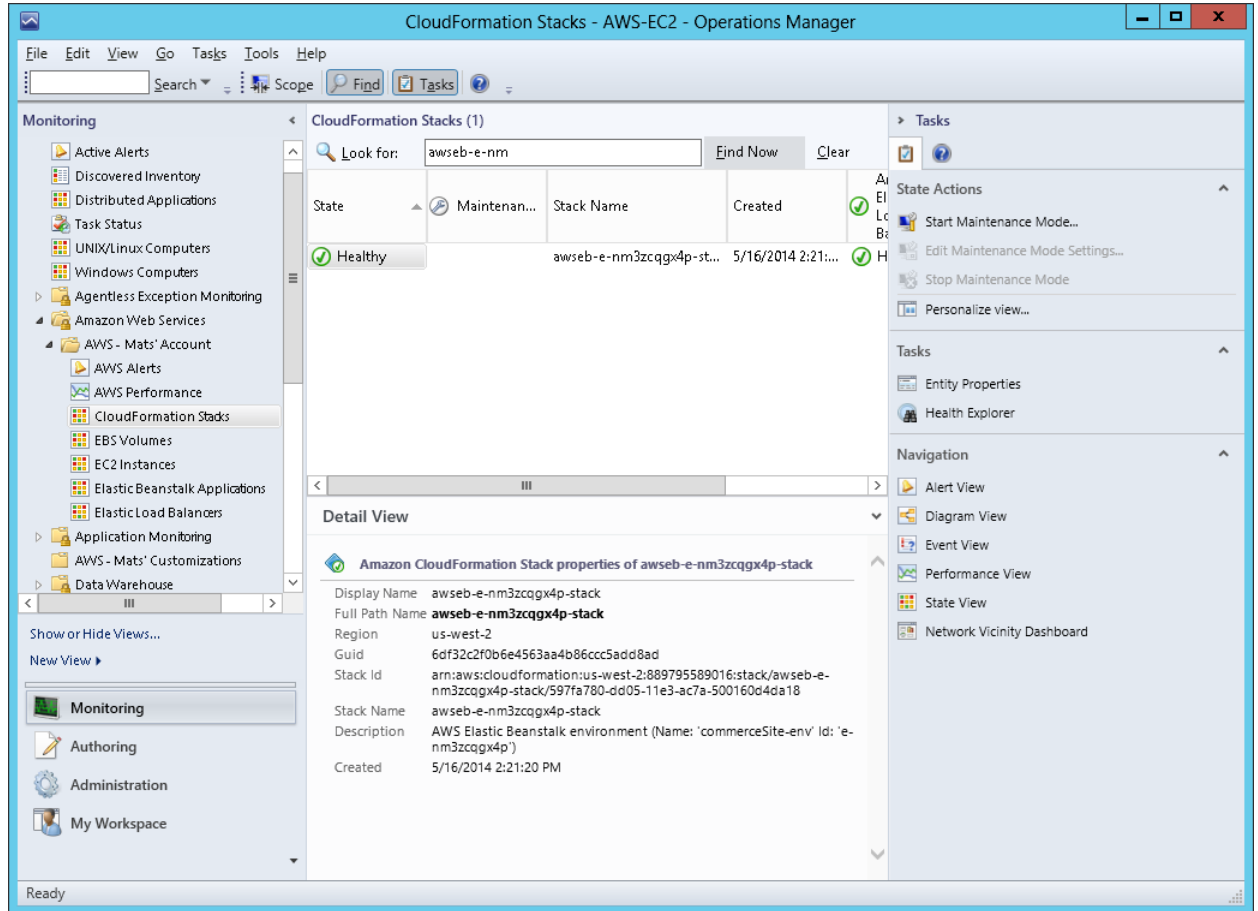
Amazon Elastic Compute Cloud User Guide for Microsoft Windows Views



CloudFormation Stacks

Shows the health state of all the AWS CloudFormation stacks for a particular AWS account from all regions.

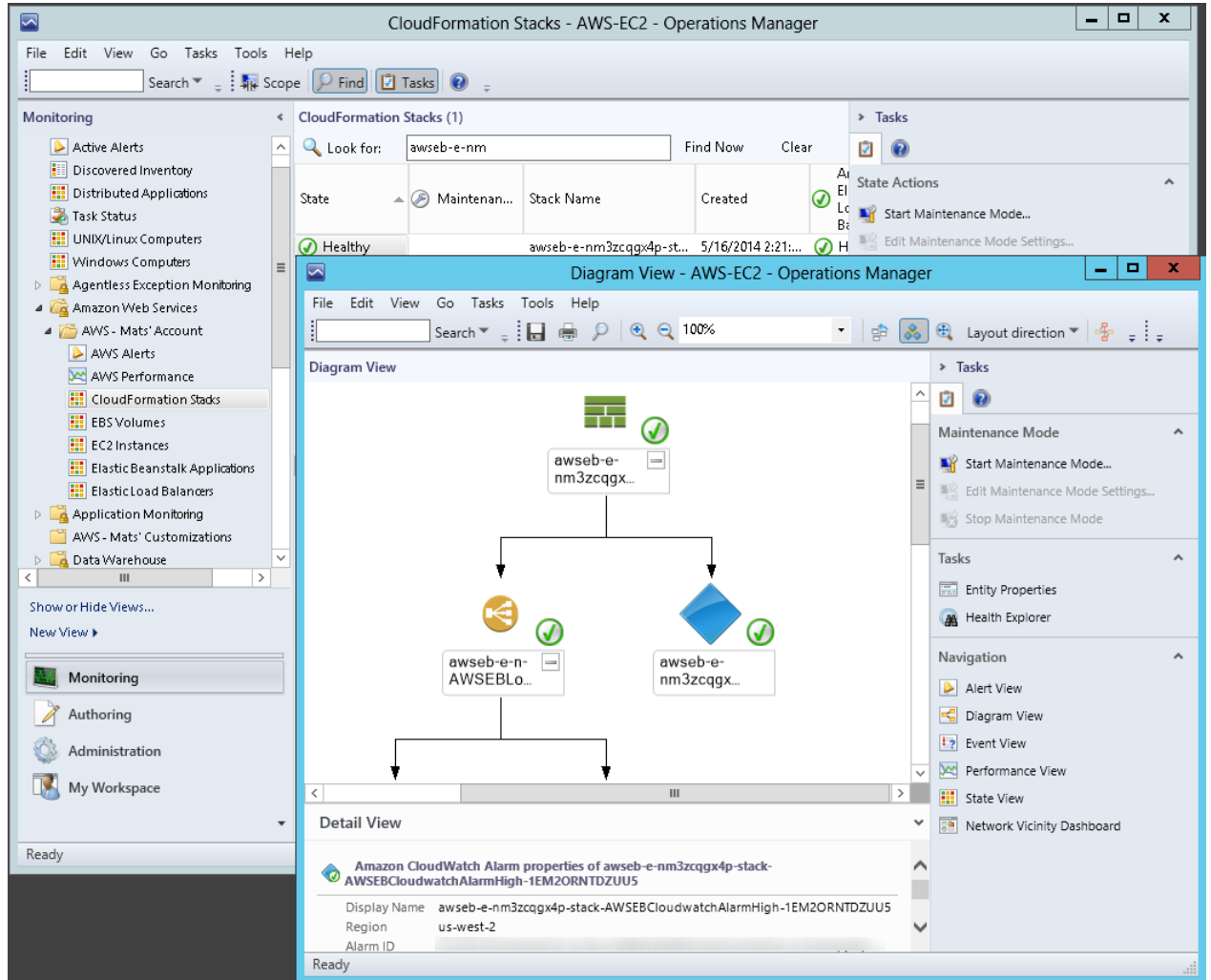
Amazon Elastic Compute Cloud User Guide for Microsoft Windows Views



CloudFormation Stacks Diagram View

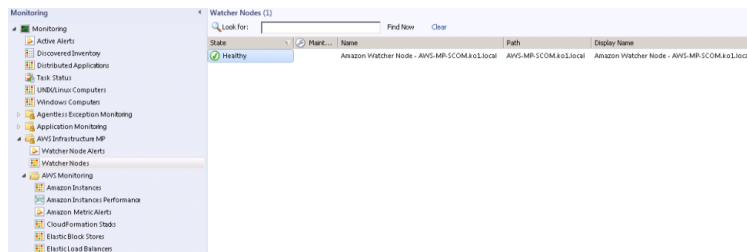
Shows the AWS CloudFormation stack relationship with other components. An AWS CloudFormation stack might contain Amazon EC2 or Elastic Load Balancing resources. The following illustration shows an example:

Amazon Elastic Compute Cloud User Guide for Microsoft Windows Discoveries



Watcher Nodes (System Center Operations Manager 2007)

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:

- EC2 instances

Amazon Elastic Compute Cloud User Guide for Microsoft Windows Discoveries

- EBS volumes
- ELB load balancers
- AWS CloudFormation stacks
- Amazon CloudWatch metrics (default metrics for the discovered Amazon EC2, Amazon EBS, and Elastic Load Balancing resources)
- Amazon CloudWatch alarms (defined for the discovered metrics)
- AWS Elastic Beanstalk applications
- Auto Scaling groups and Availability Zones

For Amazon CloudWatch metrics discovery, the following guidelines apply:

- Amazon CloudWatch metrics in the diagram views appear as **Not Monitored** if no Amazon CloudWatch alarms are defined for that metric.
- Only default Amazon CloudWatch metrics appear in Operations Manager. Custom Amazon CloudWatch metrics do not appear in Operations Manager.
- 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.

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
- EC2 instance and its EBS volumes
- 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.

Discovery	Runs On	Interval (seconds)
Watcher Node Discovery Targets the root management server and creates the watcher node objects.	Management server	14400

Discovery	Runs On	Interval (seconds)
<p>Unix and Windows Computer Discovery</p> <p>Finds Unix and Windows computers that are running on EC2 instances. As a result, a simple URL-querying script is executed on the computers to identify the EC2 instance ID that can be used for linking EC2 instance objects to Unix and Windows computers. This discovery populates the properties of the <code>AmazonComputerLink</code> objects.</p>	<p>Unix or Windows computer</p>	<p>14400</p>
<p>EC2 Instance to Unix or Windows Computer Relation Discovery</p> <p>Discovers the relationship between the EC2 instance and the Unix or Windows computer.</p>	<p>Management server</p>	<p>14400</p>
<p>AWS Elastic Beanstalk Discovery</p> <p>Discovers AWS Elastic Beanstalk and its relationship with environment, resources, and configuration.</p>	<p>Management server (System Center 2012) Watcher node (System Center 2007 R2)</p>	<p>14400</p>

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).

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Rules**

Monitor	Interval (seconds)
AWS CloudFormation Stack Status	900
Amazon CloudWatch Metric Alarm	900
Amazon EBS Volume Status	900
Amazon EC2 Instance Status	900
Amazon EC2 Instance System Status	900
Watcher Node to Amazon Cloud Connectivity	900

Rules

Rules create alerts (based on Amazon CloudWatch metrics) and collect data for analysis and reporting.

Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Rules

Interval (seconds)

SWA4400

cross

yesD

elUR

stgA

eh t

recta

edon

dna

sesu

eh t

SWA

I PA

o t

resid

stejo

rof

eh t

g/bf

SWA

sur

2CE

asi

SBE

slv

do l

sub

dna

SWA

ri/c

sas

td/c

scir

r o

ra la

e r a

t on

pad

ret A

yesid

s i

can

ve i v

eh t

stejo

n i

eh t

td

edid

das

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Rules**

Interval (seconds)

noza4400
 lozC
 scir
 dna
 maA
 yesD
 eluR

steT
 eh t
 stejo
 rof
 etla
 SWA
 ser

dna
 sesid
 eh t
 tlel
 nozaA
 lozC
 scir

dna
 ,sala
 f i
 ,ya
 daca
 ht iw
 esht
 .sine

noza00
 ci tE
 koIB
 erotS
 mló/
 enP
 scir
 ataD
 rit6
 eluR

noza00
 2CE
 cas
 enP
 scir
 ataD
 rit6
 eluR

Interval (seconds)
300
60
30
15
5
1
0.5
0.1

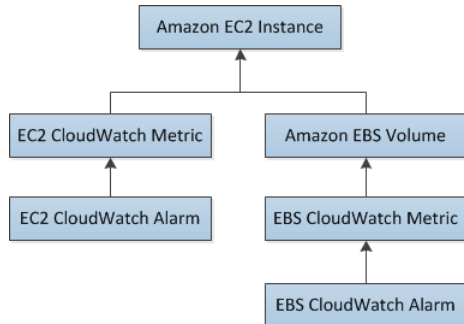
Events

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	AWS Elastic Beanstalk Discovery finished
4116	AWS Elastic Beanstalk Environment Status State calculated
4117	AWS Elastic Beanstalk Environment Operational State calculated
4118	AWS Elastic Beanstalk Environment Configuration State calculated

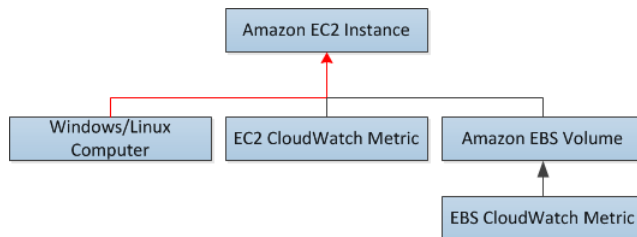
Health Model

The following illustration shows how the health states roll up in the AWS Management Pack.



The health state for an Amazon CloudWatch alarm rolls up to the corresponding Amazon CloudWatch metric. So the Amazon CloudWatch metrics for Amazon EC2 roll up their health state to the Amazon EC2 instance. Similarly, the Amazon CloudWatch metrics for Amazon EBS roll up their health state to the Amazon EBS volume. The Amazon EBS volumes used by an Amazon EC2 instance roll up their health state to the Amazon EC2 instance.

When the relationship between an Amazon EC2 instance and its operating system has been discovered, the operating system health state rolls up to the Amazon EC2 instance.



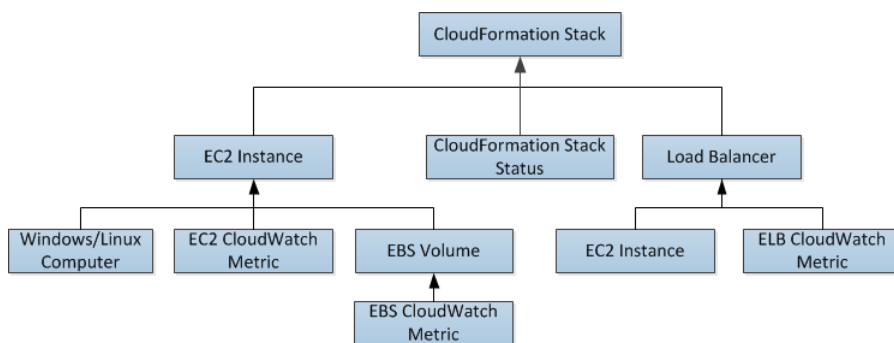
The health state of an AWS CloudFormation stack depends on the status of the AWS CloudFormation stack itself and the health states of its resources, namely the Elastic Load Balancing load balancers and Amazon EC2 instances.

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
Warning	UPDATE_ROLLBACK_IN_PROGRESS UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS UPDATE_ROLLBACK_COMPLETE	Recovering after some problem

Health State	AWS CloudFormation Stack Status	Notes
Healthy	CREATE_COMPLETE UPDATE_IN_PROGRESS UPDATE_COMPLETE_CLEANUP_IN_PROGRESS UPDATE_COMPLETE	Usable

The full health roll up model for an AWS CloudFormation stack is as follows:



Customizing the AWS Management Pack

For information about creating overrides, see [Tuning Monitoring by Using Targeting and Overrides](#) at 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](#) at the *Microsoft TechNet* website.

Troubleshooting the AWS Management Pack

If you run into trouble, try the following to resolve the issue.

- 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 \(p. 453\)](#).
- Verify that you have waited long enough for the AWS resources to be discovered (10–20 minutes).
- [System Center 2012] 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.
- [System Center 2007 R2] Verify that the watcher node is configured properly.
 - The proxy agent is enabled. For more information, see [Step 2: Configuring the Watcher Node \(p. 455\)](#).
 - 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.
- [System Center 2007 R2] Verify that the watcher node is healthy and resolve all alerts. For more information, see [Views \(p. 462\)](#).
- 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 \(p. 478\)](#) 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:

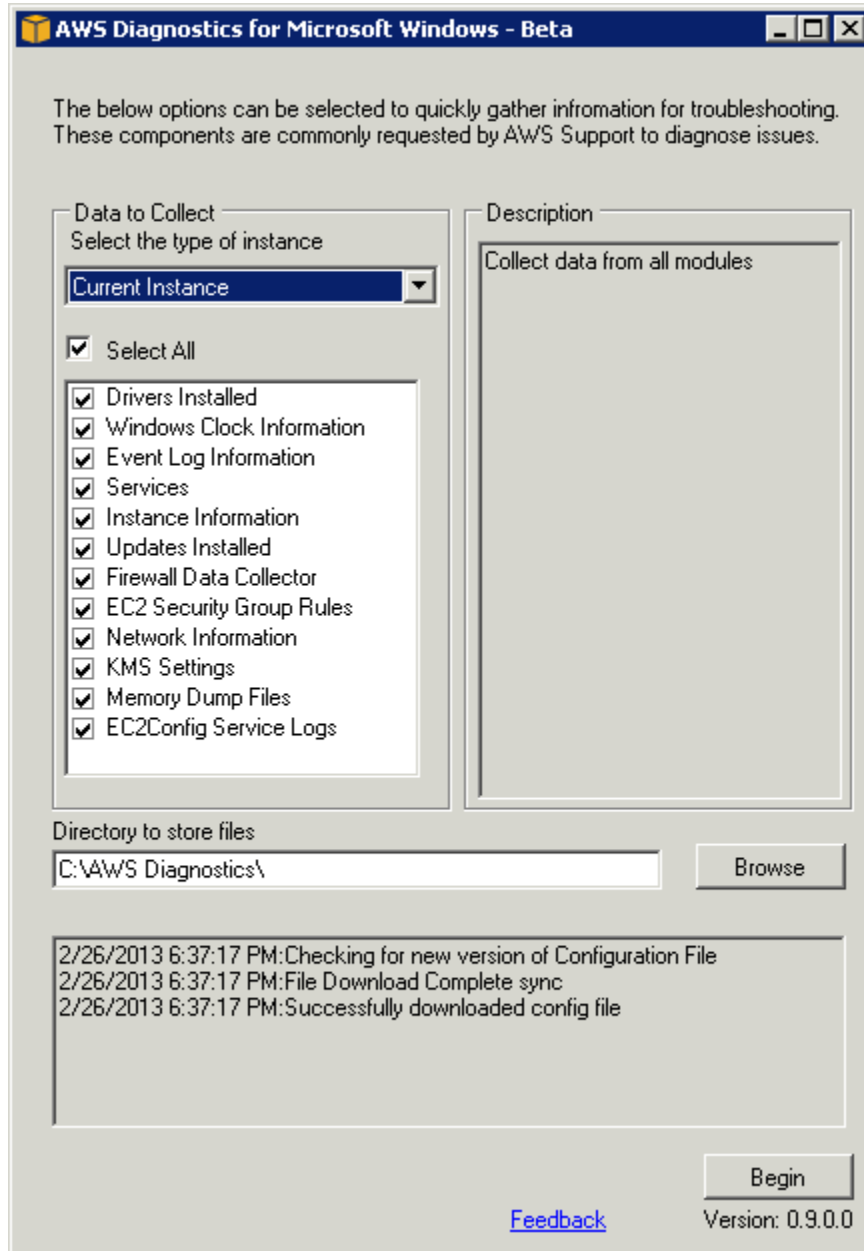
- Check for activation status and KMS settings
- Check for proper route table entries for metadata and KMS access
- Compare security group rules with Windows Firewall rules
- Check the version of the PV driver (RedHat or Citrix)
- Check whether the `RealTimeIsUniversal` registry key is set
- Check the 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.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Analyzing the Current Instance**



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 dir-

ectory 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 are common issues and error messages that you might encounter when you start or connect to your Windows instance.

Common Issues and Messages

- [No console output](#) (p. 487)
- [Instance terminates immediately](#) (p. 488)
- ["Password is not available"](#) (p. 488)
- ["Password not available yet"](#) (p. 489)
- ["Cannot retrieve Windows password"](#) (p. 489)
- ["Waiting for the metadata service"](#) (p. 489)
- [Remote Desktop can't connect to the remote computer](#) (p. 492)
- [RDP displays a black screen instead of the desktop](#) (p. 494)
- ["Unable to activate Windows"](#) (p. 495)
- ["Windows is not genuine \(0x80070005\)"](#) (p. 495)
- ["No Terminal Server License Servers available to provide a license"](#) (p. 496)
- [Instance loses network connectivity or scheduled tasks don't run when expected](#) (p. 496)

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. 482).

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. 153).

To get the console output for your instance using the Amazon EC2 console, select the instance, click **Actions**, 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. 162).

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, the not 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. 175).

"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. 487).

"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. 100).

If the instance is failing the instance reachability test, it's possible that the system has been configured with a static IP address. Try the following to resolve this issue.

- [EC2-VPC] [Create a network interface \(p. 338\)](#) and [attach it to the instance \(p. 341\)](#).
- [EC2-Classic] Enable DHCP.

To enable DHCP 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 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 EC2Config from [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.

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
"Waiting for the metadata service"**

- 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 \(p. 162\)](#).
- 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 \(p. 164\)](#).
- 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 Virtual Private Cloud 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 [Monitoring Instances with Status Checks \(p. 189\)](#) and [Troubleshooting Instances with Failed Status Checks \(Amazon Elastic Compute Cloud User Guide for Linux\)](#).
- [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 Virtual Private Cloud 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 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 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

**Amazon Elastic Compute Cloud User Guide for Microsoft
Windows
Remote Desktop can't connect to the remote computer**

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 `xxxxProfile`, 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 \(p. 175\)](#).
 - 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**, 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 \(p. 97\)](#).

"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. 162).
- 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. 495).

"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 feature.
- Licensing has expired. If the licensing has expired, you can't connect to your Windows instance. You can 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 \(p. 164\)](#).
2. Verify that the following registry key exists and is set to 1: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal**

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: 2014-06-15.

Feature	API Version	Description	Release Date
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. 151) . 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. 76) .	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

Feature	API Version	Description	Release Date
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. 355) .	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. 166) .	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. 449) .	22 May 2014
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. 387) .	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 Instance Type Details .	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. 437) .	28 January 2014
Additional M3 instances	2013-10-15	The M3 instance sizes <code>m3.medium</code> and <code>m3.large</code> are now supported. For more information about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details .	20 January 2014

Feature	API Version	Description	Release Date
I2 instances	2013-10-15	These instances provide very high IOPS. I2 instances also support enhanced networking that delivers improved inter-instance latencies, lower network jitter, and significantly higher packet per second (PPS) performance. For more information, see I2 Instances (p. 79) .	19 December 2013
Updated M3 instances	2013-10-15	The M3 instance sizes, <code>m3.xlarge</code> and <code>m3.2xlarge</code> now support instance store with SSD volumes. For more information about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details .	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. 272) .	20 November 2013
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 Instance Type Details .	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. 136) .	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 GPU Instances (p. 85) .	4 November 2013
New launch wizard		There is a new and redesigned EC2 launch wizard. For more information, see Launching an Instance (p. 130) .	10 October 2013
Modifying Amazon EC2 Reserved Instances	2013-08-15	You can now modify Reserved Instances in a region.	11 September 2013

Feature	API Version	Description	Release Date
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. 324) .	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. 274) .	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. 384) .	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. 449) .	8 May 2013
New Tags page		There is a new Tags page in the Amazon EC2 console. For more information, see Tagging Your Amazon EC2 Resources (p. 427) .	04 April 2013
Additional EBS-optimized instance types	2013-02-01	The following instance types can now be launched as EBS-optimized instances: <code>c1.xlarge</code> , <code>m2.2xlarge</code> , <code>m3.xlarge</code> , and <code>m3.2xlarge</code> . For more information, see Amazon EBS–Optimized Instances (p. 93) .	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. 164) .	March 2013
AWS Diagnostics for Microsoft Windows Server		The topic AWS Diagnostics for Microsoft Windows Server - Beta (p. 482) 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. 67) .	11 March 2013

Feature	API Version	Description	Release Date
Launch instances into a default VPC	2013-02-01	<p>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.</p> <p>For more information, see Supported Platforms (p. 312).</p>	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. 82) .	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. 384) .	17 December 2012
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. 365) . Also added new status checks for Provisioned IOPS (SSD) volumes. For more information, see Monitoring Volumes with Status Checks (p. 368) .	20 November 2012

Feature	API Version	Description	Release Date
Support for Microsoft Windows Server 2012		<p>Amazon EC2 now provides you with several pre-configured 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:</p> <ul style="list-style-type: none"> • 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 Instance Type Details .	31 October 2012
Amazon EC2 Spot Instance Request Status	2012-10-01	Spot Instance request status makes it easy to determine the state of your Amazon EC2 Spot requests.	14 October 2012
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

Feature	API Version	Description	Release Date
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. 355) .	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 H1 Instances (p. 81) .	18 July 2012
IAM roles on Amazon EC2 instances	2012-06-01	IAM roles for Amazon EC2 provide: <ul style="list-style-type: none"> • 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.		Using Auto Scaling, you can now manage your Spot Instances: <ul style="list-style-type: none"> • Place bids for Amazon EC2 Spot Instances using Auto Scaling launch configurations, and set up a schedule for placing bids for Spot Instances. • 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
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 <code>cc2.8xlarge</code> instances in a VPC.	26 April 2012
AWS Marketplace AMIs	2012-04-01	Added support for AWS Marketplace AMIs.	19 April 2012

Feature	API Version	Description	Release Date
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. 334).	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: <i>Heavy Utilization</i> , <i>Medium Utilization</i> , and <i>Light Utilization</i> .	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. 189).	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
Amazon EC2 Spot Instances in Amazon VPC	2011-07-15	Added information about the support for Amazon EC2 Spot Instances in Amazon VPC. With this update, users will be able to launch Spot Instances in the Amazon Virtual Private Cloud (Amazon VPC). By launching Spot Instances in Amazon VPC, users of Spot Instances can enjoy all of the controls and advanced security options of Amazon VPC.	11 October 2011
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 <code>ec2-import-instance</code> and <code>ec2-import-volume</code> , which now will perform the upload of the images into Amazon EC2 after creating the import task. In addition, with the introduction of the <code>ec2-resume-import</code> 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. 113).	15 September 2011

Feature	API Version	Description	Release Date
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. 112) .	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 (p. 111) .	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. 272) .	26 April 2011
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 <i>Amazon Virtual Private Cloud User Guide</i> .	27 March 2011

Feature	API Version	Description	Release Date
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 pre-configured 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. 100) and Instance Metadata Categories (p. 103) .	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 (p. 111) .	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. 375) .	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. 147) .	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. 112) .	15 December 2010
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 Instance Type Details .	14 November 2010

Feature	API Version	Description	Release Date
Filters and Tags	2010-08-31	Added information about listing, filtering, and tagging resources. For more information, see Listing and Filtering Your Resources (p. 425) and Tagging Your Amazon EC2 Resources (p. 427) .	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 <code>t1.micro</code> instance type for certain types of applications. For more information, see T1 Micro Instances (p. 86) .	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. 272) .	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 Instance Type Details .	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. 365) .	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 Instance Type Details .	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*.