
Amazon Elastic Compute Cloud

Microsoft Windows Guide

API Version 2011-07-15



Amazon Elastic Compute Cloud: Microsoft Windows Guide

Copyright © 2011 Amazon Web Services LLC or its affiliates. All rights reserved.

Table of Contents

Welcome	1
Amazon Elastic Compute Cloud for Windows Users	3
Using Windows AMIs	11
Amazon Windows AMI Basics	11
Choosing a Windows AMI	14
Creating Your Own Windows AMI	18
Using EC2Config Service	18
Creating an Amazon EBS-Backed Windows AMI	23
Creating an Instance Store-Backed Windows AMI	25
Shared Windows AMIs	29
Paid Windows AMIs	34
Installing the Amazon EC2 Command Line Tools on Windows	38
Setting Up a Windows HPC Cluster on Amazon Elastic Compute Cloud	43
Document History	56
Glossary	53
Index	57

Welcome

This is the *Amazon Elastic Compute Cloud Microsoft Windows Guide*. This user guide contains conceptual information about the Amazon EC2 web service, as well as information about how to use the service to create new web applications on Windows instances. Separate sections describe how to program with the command line interface (CLI) and the Query API.

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable computing capacity—literally server instances in Amazon's data centers—that you use to build and host your software systems. With Amazon EC2, you can get access to infrastructure resources using APIs or web tools and utilities.

This guide picks up where the [Amazon Elastic Compute Cloud Getting Started Guide](#) leaves off.

How Do I...?

How Do I?	Relevant Sections
Get a general product overview and information about pricing	Amazon EC2 product page
Get started with Amazon EC2 Windows Instances	Amazon Elastic Compute Cloud Getting Started Guide
Get a quick summary of the basic infrastructure components that Amazon EC2 provides for Windows Instances	Introduction to Amazon EC2 (p. 3)
Get detailed information on how to use Windows AMI	Using Windows AMIs (p. 11)
Get detailed information on how to use Amazon EC2 infrastructure components	Using Amazon EC2
Find available libraries for programmatically accessing EC2	Available Libraries
Get started with the command line tools (i.e., the EC2 API tools)	Installing the Amazon EC2 Command Line Tools on Windows (p. 38)

**Amazon Elastic Compute Cloud Microsoft Windows
Guide
How Do I...?**

How Do I?	Relevant Sections
Get started using the Query or SOAP API for EC2	Making API Requests

Amazon Elastic Compute Cloud for Windows Users

Introduction to Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) is an Amazon Web Service (AWS) you can use to access servers, software, and storage resources across the Internet in a self-service manner. With Amazon EC2 you basically rent infrastructure comprising virtual servers and/or storage devices by the hour and use them to install, run, and process your applications at any time, for as long as you need, and for any legal purpose. After your requirement is fulfilled, you can either terminate the usage of the entire infrastructure or reduce the capacity and keep it in maintenance mode until you need to scale it up again. You pay for only what you use, and there is no minimum charge. With Amazon EC2 you do not need to invest in expensive hardware and have it sitting idle when your traffic or compute requirement is low. For a hands-on introduction to Amazon EC2, go to the [Amazon Elastic Cloud Compute User Guide](#).

Amazon EC2 and Windows

How does Amazon EC2 work with your Windows environment? Amazon EC2 provides templates known as Amazon Machine Images (AMIs) that contain pre-configured software such as an operating system, application server, and applications. You use these templates to launch your server instances, which are running copies of the AMI. After you launch your instance, you use it just like a physical server. You can also launch multiple instances of an AMI, thus replicating the same configuration across each of the instances. Amazon publishes a large selection of AMIs that contain software configurations specific to the Windows platform. In addition, members of the AWS developer community have published their own custom Windows AMIs. You might only need to use the Windows AMIs that Amazon or other reputable sources provide, and you can simply customize the resulting Windows instances (by running a script) to provide the data or software you need each time you launch an instance. You can also create custom Windows AMIs with pre-installed and pre-configured applications. These AMIs can then be launched quickly and efficiently to become part of a live deployment. For detailed information on Amazon Windows AMIs, see [Using Windows AMIs \(p. 11\)](#) and for information on using AMIs and Instances, see [Using Amazon EC2](#).

Differences Between Windows Server and an Amazon EC2 Windows Instance

Amazon EC2 infrastructure is composed of virtual servers accessed via the Internet. These are commonly called *cloud servers*. By using Amazon EC2, you eliminate the need to buy and maintain expensive hardware. The architecture of applications running on cloud servers can differ significantly from the traditional application models, so implementing applications on cloud servers requires a fundamental shift in your design process.

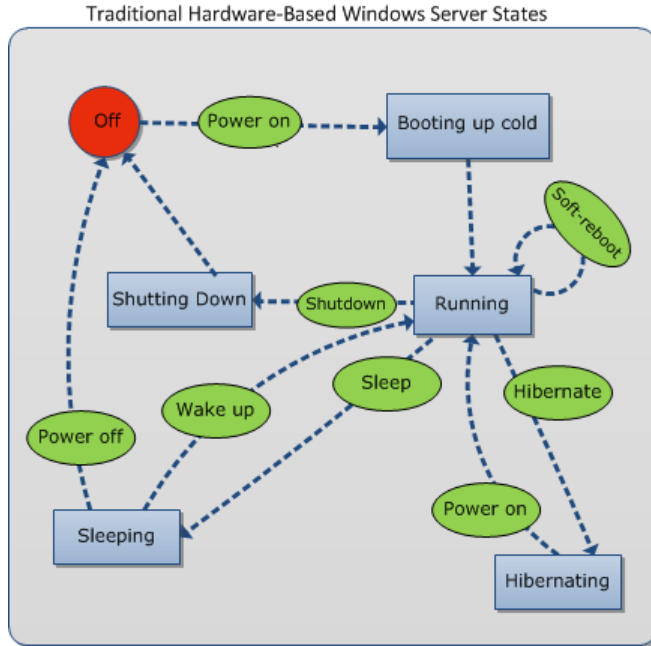
Before you begin launching Amazon EC2 Windows instances, you need to understand how the architectures of traditional hardware-based infrastructure and cloud-based infrastructure differ. The following table describes some key differences between a Windows Server and an Amazon EC2 Windows instance.

Amazon EC2 Windows Instance	Windows Server
Designed to be deployed and terminated on demand.	Cannot be easily discarded after it is setup.
Resources and capacity are scalable.	Resources and capacity are physically limited.
You pay for the usage of the infrastructure. Billing stops as soon as the instance is terminated.	You pay for the infrastructure, whether you use it or not.
Does not occupy physical space and does not require regular maintenance.	Occupies physical space and has to be maintained on a regular basis.

After you launch your Amazon EC2 Windows instance, it behaves a lot like a traditional hardware-based Windows Server. For example, both a 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 compute model. The way an Amazon EC2 instance runs is not the same as the way a traditional Windows Server runs.

A traditional Windows Server goes through a number of phases from the time it is booted up through the time it is shut down as the following diagram shows.

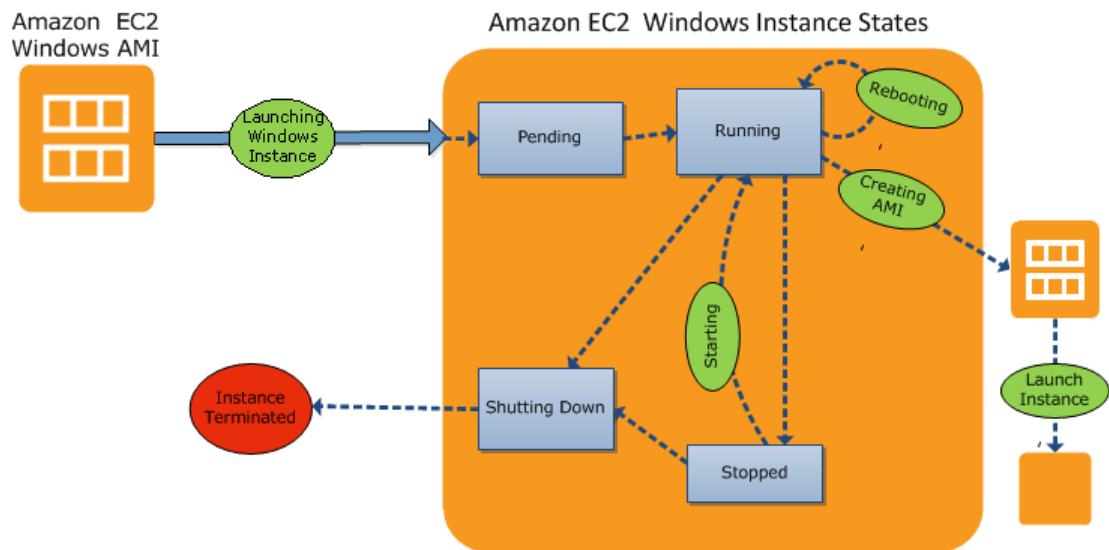
**Amazon Elastic Compute Cloud Microsoft Windows
Guide
Differences Between Windows Server and an Amazon
EC2 Windows Instance**



A traditional hardware-based Windows Server starts with a push of a power button. This is called *cold booting*. When the server is up and running, you can choose to either keep the server running until it is time to shut it down, keep it in a sleep state for a specific time duration, or keep it in a state of hibernation. The server is powered down during the hibernating and sleep states. These states can be brought back to the running state by powering the Windows Server on. However, once the server is powered off, the only way to get it up and running is by cold booting.

When your traditional Windows Server is powered off, all the resources associated with that server remain intact and in the state they were in when you switched it off. The information you stored on the hard drives persists and is ready to be accessed whenever needed.

An Amazon EC2 Windows instance has a number of similarities with the traditional hardware-based server, as you can see by comparing the following diagram with the previous diagram.



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

An Amazon EC2 Windows instance starts with the launch of the instance. Next, it briefly goes into the pending state while registration takes place. Then it moves to the running state, where instances can be rebooted, stopped, and then re-started. The Windows instance remains active until you initiate a shutdown process that terminates the instance. You can create an image of your instance and launch additional instances while your Amazon EC2 Windows instance is in the running state. This feature allows you to scale your infrastructure on demand.



Note

After an Amazon EC2 Windows instance is terminated, its infrastructure is no longer available to you. If you want to continue working with the same infrastructure, you will have to launch a new instance.

You have control over Amazon EC2 instances and the resources that come with them, as long as they are in running or in stopped states. After the instance is terminated, you can choose to launch another instance of the same configuration, or a different configuration that meets a different requirement.

Designing Your Applications to Run on Amazon EC2 Windows Instances

It is extremely 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 compute) 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 compute, 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 each other 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 will improve high availability in your system. For more information, see [Using Regions and Availability Zones](#).

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

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

EC2 environment, you should encrypt the data before uploading it. On Amazon EC2, file encryption depends on the operating system.

Basic Infrastructure Components of Amazon EC2

You might be considering creating a new application to run on Amazon EC2, or moving an existing application from your own servers into Amazon EC2 instances. To do either, you should understand the Amazon EC2 infrastructure components and how they are similar to or different from your own data centers. This section gives a brief description of the main components that Amazon EC2 provides.

Amazon Machine Images and Instances

An *Amazon Machine Image (AMI)* is a template that contains a software configuration (e.g., operating system, application server, and applications). From an AMI, you launch *instances*, which are running copies of the AMI. You can launch multiple instances of an AMI. For more information on using Windows AMIs, see [Using Windows AMIs \(p. 11\)](#).

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.

You can use a single AMI or multiple AMIs depending on your needs. From a single AMI, you can launch different types of instances. An *instance type* is essentially a hardware archetype with differing compute and memory facilities. You select a particular instance type based on the amount of memory and computing power you need for the application or software that you plan to run on the instance. For more information about the available Windows instance types, see [Windows Instance Types](#).

Regions and Availability Zones

Amazon has data centers in different areas of the world (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, go to the [Amazon EC2 Pricing page](#)).

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](#).

Storage

When using Amazon EC2, you might have data that you need to store. The two most commonly used storage types are:

- [Amazon Simple Storage Service \(Amazon S3\)](#)
- [Amazon Elastic Block Store \(Amazon EBS\) volumes](#)

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, go to the [Amazon S3 product page](#).

Amazon EBS Volumes

Amazon EBS provides your instances with persistent, block-level storage. Amazon EBS volumes are essentially the equivalent of physical hard disks that you can attach to a running instance.

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

You can attach multiple volumes to an instance. To keep a back-up copy, you can create a [snapshot](#) of the volume that 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 one. For more information about Amazon EBS volumes, see [Amazon Elastic Block Store](#).

Instance Store

Some EC2 instance types comes with a preconfigured block of pre-attached disk storage. This is called an [instance store](#); it is also known as an ephemeral store. An instance store is dedicated to a particular instance, and it provides temporary block-level storage. The data on the instance store volumes persists only during the life of the associated Amazon EC2 instance. The amount of this storage ranges from 160 GiB to up to 1.7 TiB and varies by Amazon EC2 instance type. Larger Amazon EC2 instances have more instance volumes and of larger sizes. For more information, see [Amazon EC2 Instance Storage](#).

Root Device Storage

When Amazon EC2 was first introduced, all AMIs were *backed by the Amazon EC2 instance store*, which means that the root device for an instance launched from the AMI is stored in the instance store. After we introduced Amazon EBS, we also introduced AMIs that are *backed by Amazon EBS*, which means that the root device for an instance launched from the AMI is an Amazon EBS volume. At present we provide support for instance store-backed AMIs only on Windows Server 2003. All Amazon EC2 Windows AMIs on Windows Server 2008 and later are backed by Amazon EBS. The description of an AMI includes which type it is (you'll see the root device referred to in some places as either *ebs* (for Amazon EBS-backed) or *instance-store* (for Amazon instance store-backed)). For more information, see [Root Device Storage on Windows AMIs \(p. 17\)](#)

Networking and Security

Each instance is launched into the Amazon EC2 network space and assigned a public IP address. Instances can fail or terminate for reasons outside of your control. If one fails and you launch a replacement instance, the replacement will have 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 [Using Instance IP Addresses](#).

You use *security groups* to control who can access your instances. These are analogous to an inbound network firewall that allows 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 in to 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 about security groups, see [Using security Groups](#).

Monitoring, Auto Scaling, and Load Balancing

AWS provides several features that enable you to do the following:

- Monitor basic statistics for your instances and Amazon EBS volumes.
For more information, see [CloudWatch Developer Guide](#).
- Automatically scale your Amazon EC2 capacity up or down according to conditions you define.
For more information, go to the [Auto Scaling Developer Guide](#).
- Automatically distribute incoming application traffic across multiple Amazon EC2 instances.
For more information, go to the [Elastic Load Balancing Developer Guide](#).

AWS Identity and Access Management

Amazon EC2 integrates with AWS Identity and Access Management (IAM), a service that lets your organization do the following:

- Create users and groups under your organization's AWS account
- Easily share your AWS account resources between the users in the account
- Assign unique security credentials to each user
- Granularly control user's access to services and resources
- Get a single AWS bill for all users under the AWS account

For example, you can use IAM with Amazon EC2 to control which users under your AWS account can create AMIs or launch instances. For more information, go to [Using AWS Identity and Access Management](#).

Available Interfaces

AWS provides different interfaces to access EC2.

- **AWS Management Console**—The AWS Management Console is a simple web-based GUI. For more information about using the console, go to the [Amazon Elastic Compute Cloud Getting Started Guide](#).
- **Command Line Tools (API Tools)**—EC2 provides a Java-based command-line client that wraps the EC2 SOAP API. For more information on installing command line tools, see [Installing the Amazon EC2 Command Line Tools on Windows \(p. 38\)](#).
- **Programmatic Interface**—The following table lists how you can access EC2 programmatically.

Type of Access	Description
AWS SDKs	AWS provides the following SDKs: <ul style="list-style-type: none">• AWS SDK for Java• AWS SDK for .NET• AWS SDK for PHP• AWS SDK for Ruby

**Amazon Elastic Compute Cloud Microsoft Windows
Guide
How You're Charged for Amazon EC2**

Type of Access	Description
Third-Party Libraries	Developers in the AWS developer community also provide their own libraries, which you can find at the following AWS developer centers: <ul style="list-style-type: none">• AWS Java Developer Center• AWS PHP Developer Center• AWS Python Developer Center• AWS Ruby Developer Center• AWS Windows and .NET Developer Center
EC2 API	If you prefer, you can code directly to the EC2 API (Query or SOAP). For more information, see Making API Requests , and go to Amazon Elastic Compute Cloud API Reference .

How You're Charged for Amazon EC2

With Amazon EC2, you pay for only what you use, and there's no minimum charge. Your charges are broken down into these general parts:

- Instance usage
- Data transfer
- Storage

For a complete list of charges and specific prices, go to the [Amazon EC2 pricing page](#). To calculate the cost of a sample provisioned environment, go to [AWS Economics Center](#) and use [Amazon EC2 Cost Comparison Calculator](#).

To see your bill, go to [AWS Account Activity page](#).

Learn More about Amazon EC2

We recommend you understand how the infrastructure components described briefly in this section work before designing your application or service to run on Amazon EC2. For detailed information on Amazon EC2 infrastructure components, go to [Using Amazon EC2](#).

Using Windows AMIs

An Amazon Machine Image (AMI) is the image of the machine that contains all the information necessary to boot instances of your software. It is somewhat similar to a snapshot of the boot partition containing the operating system and installed software running on your server. You use these images to launch your Amazon EC2 Windows instances, which are running copies of the AMI.

For more information on Amazon Windows AMIs, see [Amazon Windows AMI Basics \(p. 11\)](#).

You can use the [AWS Management Console](#) to search for Windows AMIs that meet specific criteria, and then launch instances of those AMIs. For example, you can view the Windows AMIs provided by Amazon, or the Windows AMIs provided by the EC2 community. For more information on choosing a Windows AMI, see [Choosing a Windows AMI \(p. 14\)](#).

You might find public AMIs that suit your needs. You can customize a public AMI and then save that customized AMI for your own use and create a new AMI. For more information see [Creating Your Own Windows AMI \(p. 18\)](#).

After you create a new AMI, you can keep it private so that only you can use it, or you can share it with other AWS accounts that you specify. You can also make your customized AMI public so that the EC2 community can use it. Building safe, secure, usable AMIs for public consumption is a fairly straightforward process, if you follow a few simple guidelines. For information on how to create and use shared AMIs, see [Shared Windows AMIs \(p. 29\)](#).

Paid AMIs are AMIs that you purchase from third parties or AMIs that come with service contracts from organizations such as Red Hat. If you're interested in selling an AMI to other developers, see [Amazon DevPay](#). You can also create your AMIs and sell it to other EC2 users. For more information on selling or using paid AMIs, see [Paid Windows AMIs \(p. 34\)](#).

To help categorize and manage your AMIs, you can assign *tags* of your choice to them. For more information, go to [Using Tags](#).

Amazon Windows AMI Basics

Amazon Web Services (AWS) provides a set of publicly available AMIs that contain software configurations specific to the Windows platform, so that you can quickly start building and deploying your applications using Amazon EC2. You first choose the AMI that meets your specific requirements, launch an EC2 instance (virtual server) off of that AMI, connect to the instance just as you would connect to a virtual server, and then use that EC2 instance just as you would use a traditional hardware-based Windows server.

Amazon Elastic Compute Cloud Microsoft Windows Guide Amazon Windows AMI Basics

Amazon AWS currently provides the following basic versions of Windows AMIs.

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

AWS also provides a set of publicly available AMIs that are pre-bundled with 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 Amazon Windows AMI pre-bundled with SQL Server Express, IIS, and ASP.NET to launch an instance that runs web and ASP.NET applications. Launching an instance from an Amazon Windows AMIs 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 AMI, customize the instance by installing 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.

In addition to the public AMIs provided by AWS, there are AMIs published by the AWS developer community available for your use. We highly recommend that you use only those Windows AMIs that Amazon or other reputable sources provide.

For a list of Amazon-approved Microsoft Windows AMIs, go to [Amazon Machine Images \(AMIs\)](#) on the AWS website. You can refine your search by mentioning the platform name *Microsoft Windows* in the search box as shown in the following screenshot.

The screenshot shows the AWS Management Console interface. At the top, there is a search bar with the text 'AMIs' and a dropdown menu showing 'Microsoft Windows' selected. Below the search bar, there are navigation tabs for 'AWS', 'Products', 'Developers', 'Community', 'Support', and 'Account'. The main content area displays 'Amazon Machine Images (AMIs)' with a breadcrumb trail: '"Microsoft Windows" > Amazon Machine Images (AMIs)'. Below this, it shows 'Showing 1-25 of 101 results' and a 'Sort by:' dropdown. The first two results are listed:

- Microsoft Windows Server 2008 R2 with SQL Server Standard 2008 R2**
Amazon Machine Images (AMIs)
This AMI, published by Amazon, contains an installation of Microsoft Windows Server 2008 R2 plus SQL Server Standard 2008 R2 pre-installed in the system.
- Microsoft Windows Server 2008 R2 with IIS and SQL Server Express 2008 R2**
Amazon Machine Images (AMIs)
This AMI, published by Amazon, contains an installation of Microsoft Windows Server 2008 R2 , with IIS and SQL Server Express 2008 R2 pre-installed in the system.

On the left side, there is a 'Refine Your Search' sidebar with filters for 'Operating System' (Linux/Unix (9), Microsoft Windows (40), Other (3), Windows (49)) and 'Region' (AP Southeast (43), Europe (65), US East (99), US West (39)). At the bottom left, there is a 'Developer Resources' section.

Click any AMI in the list to see all the relevant information about it.

Configuration of an Amazon Windows AMI

The Amazon-provided 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 as a local system and is primarily used during the initial setup. EC2Config performs the following tasks when launching your instance:

- Sets the hostname to the private DNS name
- Generates and sets a random initial password on the administrator's account
- Initializes and formats all the drives attached to the instance
- Generates and installs the host certificate for Terminal Services
- Syncs the instance clock with a time server

After you launch your Windows instance with the 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. The instances launched from the customized AMI will then be launched with the new configuration. The binaries for the EC2Config Service as well as additional tools needed to configure the new Windows AMI are contained in the `%ProgramFiles%\Amazon` directory on 32-bit instances and in the `%ProgramFiles(x86)\Amazon` directory on 64-bit instances. For more information, see [Creating Your Own Windows AMI](#) (p. 18).

Xen Drivers

Amazon Windows AMIs contain a set of drivers to permit access to Xen virtualized hardware. These drivers are used by Amazon EC2 to map the instance store and Amazon Elastic Block Store (Amazon EBS) volumes to the devices. The source files for the drivers are in the `%ProgramFiles%\RedHat` directory on the 32-bit instances and on `%ProgramFiles(x86)\RedHat` directory on 64-bit instances. The two drivers are the `rhelnet`, the RedHat Paravirtualized network driver, and `rhelscsi`, the RedHat SCSI miniport driver.

Keeping Your Instances Updated

The Amazon Windows AMIs provided at the initial launch of your Windows instances contain all the latest security updates. However, once you launch your instance, you are responsible for managing future updates, including the updates issued after the AMI was built. 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 appropriate for that software. We recommend that you first run the Windows Update service with every new Windows instance that you launch.



Note

An Amazon EC2 Windows instance can be rebooted after the updates take place. Rebooting works the same way for both *instance store-backed* instances and *Amazon EBS-backed instances*. For more information, see [Root Device Storage on Windows AMIs](#) (p. 17).

Support

Support for installation and use of the base Amazon Windows AMI is included through subscriptions to AWS Premium Support. For more information, go to [Premium Support](#).

You're encouraged to post any questions you have on using Amazon Windows AMIs to the [Amazon EC2 forums](#).

You can report issues either to Premium Support or the Amazon EC2 forums.

Choosing a Windows AMI

Amazon Machine Images (AMIs) are the basic building block of Amazon EC2. Before you accomplish anything with Amazon EC2, you must first choose the AMI that you want to work on. The AMI can either be provided by Amazon, or provided by the Amazon EC2 community, or created by you. However, if you want to create your own AMI, you must start by using one of the base AMIs provided.

Using the AWS Management Console

To view a list of available AMIs

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. In the Amazon EC2 console, click **AMIs** in the **Navigation** pane.
The console displays a list of all available AMIs. Use the **Viewing** options to narrow the list of displayed AMIs.
3. To see a list of all Windows AMIs provided by Amazon, select **Amazon Images** from the **Viewing** drop-down list.
4. Select **Windows** from the list.
5. In the list of Windows AMIs, click an AMI to view its properties. The AMI properties appear in the **Description** tab in the lower pane.

Using Command Line Tools

Amazon EC2 provides a Java-based command-line client that wraps the EC2 SOAP API. You must have installed the command line tool on your machine to use the Java-based command-line client. For information on installing command line tools see [Installing the Amazon EC2 Command Line Tools on Windows \(p. 38\)](#).

To find a suitable AMI

- Use the **ec2-describe-images** command.



Tip

You can filter the list to return only certain types of AMIs of interest to you. For more information about how to filter the results, go to [ec2-describe-images](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

The following example shows the command and the resulting output consisting of a partial list of all Amazon Windows AMIs.

**Amazon Elastic Compute Cloud Microsoft Windows
Guide
Using Command Line Tools**

```
C:\> ec2-describe-images -o amazon --filter "platform=windows"

IMAGE ami-de24d6b7    amazon/Windows-2008R2-SP1-Base-Locale-JA-JP-v101
amazon available public x86_64 machine windows
ebs hvm xen BLOCKDEVICEMAPPING /dev/sda1 snap-
3e2c5e52 35
IMAGE ami-d024d6b9    amazon/Windows-2008R2-SP1-Base-Locale-KO-KR-v101
amazon available public x86_64 machine windows
ebs hvm xen BLOCKDEVICEMAPPING /dev/sda1 snap-
022c5e6e 35
IMAGE ami-b424d6dd    amazon/Windows-2008R2-SP1-Base-Locale-ZH-CN-v101
amazon available public x86_64 machine windows
ebs hvm xen BLOCKDEVICEMAPPING /dev/sda1 snap-
4612602a 35
IMAGE ami-b624d6df    amazon/Windows-2008R2-SP1-Base-Locale-ZH-HK-v101
amazon available public x86_64 machine windows
ebs hvm xen BLOCKDEVICEMAPPING /dev/sda1 snap-
5a126036 35
IMAGE ami-98ef1df1    amazon/Windows-2008R2-SP1-Core-v101 amazon available
public x86_64 machine windows ebs hvm
xen BLOCKDEVICEMAPPING /dev/sda1 snap-laf16b76 16
IMAGE ami-1cbd4475    amazon/Windows-2008R2-SP1-English-Base-v103
amazon available public x86_64 machine windows ebs
hvm xen BLOCKDEVICEMAPPING /dev/sda1 snap-2098704e
35
IMAGE ami-42bd442b    amazon/Windows-2008R2-SP1-English-SQLExpress-v102
amazon available public x86_64 machine windows
ebs hvm xen BLOCKDEVICEMAPPING /dev/sda1 snap-
ec896182 35
IMAGE ami-baed1fd3    amazon/Windows-2008R2-SP1-English-SQLStandard-v101
amazon available public x86_64 machine windows
ebs hvm xen BLOCKDEVICEMAPPING /dev/sda1 snap-
26de574a 35
IMAGE ami-988577f1    amazon/Windows-2008R2-SP1-MultiLang-Base-v101
amazon available public x86_64 machine windows ebs
hvm xen BLOCKDEVICEMAPPING /dev/sda1 snap-d035a2bc
35
IMAGE ami-0a8a7863    amazon/Windows-2008R2-SP1-MultiLang-SQLExpress-v101
amazon available public x86_64 machine windows
ebs hvm xen BLOCKDEVICEMAPPING /dev/sda1 snap-
12cb5d7e 35
IMAGE ami-a8e705c1    ec2-paid-ibm-images/ibm-infosphere-is-winclient.mani
fest.xml amazon available public EC129708 i386 machine
windows instance-store hvm xen
IMAGE ami-df20c3b6    ec2-public-windows-images/Server2003r2-i386-Win-
v1.07.manifest.xml amazon available public i386 machine
windows instance-store hvm xen
IMAGE ami-dd20c3b4    ec2-public-windows-images/Server2003r2-x86_64-Win-
v1.07.manifest.xml amazon available public x86_64 machine
windows instance-store hvm xen
IMAGE ami-db20c3b2    ec2-public-windows-images/SqlSvrExp2003r2-i386-Win-
v1.07.manifest.xml amazon available public i386 machine
windows instance-store hvm xen
IMAGE ami-d920c3b0    ec2-public-windows-images/SqlSvrExp2003r2-x86_64-
Win-v1.07.manifest.xml amazon available public x86_64 machine
windows instance-store hvm xen
IMAGE ami-0535d66c    ec2-public-windows-images/SqlSvrStd2003r2-x86_64-
```

```
Win-v1.07.manifest.xml amazon available public x86_64 machine
windows instance-store hvm xen
```

Using the Query API

To find a suitable AMI

- Construct the following query request, which returns all Amazon-owned Windows AMIs:

```
https://ec2.amazonaws.com/
?Action=DescribeImages
&Owner.1=amazon
&Filter.1.platform=windows

&...auth parameters...
```

Following is an example response.

```
<DescribeImagesResponse xmlns="http://ec2.amazonaws.com/doc/2011-07-15/">
  <imagesSet>
    <item>
      <imageId>ami-dde40db4</imageId>
      <imageLocation>amazon/Windows-Server2008-x86_64-SqlStandard-
v103</imageLocation>
      <imageState>available</imageState>
      <imageOwnerId>206029621532</imageOwnerId>
      <isPublic>true</isPublic>
      <architecture>x86_64</architecture>
      <imageType>machine</imageType>
      <platform>windows</platform>
      <imageOwnerAlias>amazon</imageOwnerAlias>
      <name>Windows-Server2008-x86_64-SqlStandard-v103</name>
      <description>Microsoft Windows 2008 Datacenter 64-bit, Sql 2008
Standard AMI v1.03</description>
      <rootDeviceType>ebs</rootDeviceType>
      <rootDeviceName>/dev/sda1</rootDeviceName>
      <blockDeviceMapping>
        <item>
          <deviceName>/dev/sda1</deviceName>
          <ebs>
            <snapshotId>snap-349c275c</snapshotId>
            <volumeSize>30</volumeSize>
            <deleteOnTermination>true</deleteOnTermination>
          </ebs>
        </item>
      </blockDeviceMapping>
      <virtualizationType>hvm</virtualizationType>
      <hypervisor>xen</hypervisor>
    </item>
  </imagesSet>
```

**Amazon Elastic Compute Cloud Microsoft Windows
Guide
Root Device Storage on Windows AMIs**

An Amazon EC2 Windows instance can be launched from an AMI either *backed by instance store* or *backed by an Amazon Elastic Block Store (Amazon EBS) volume*. Instances launched from an instance store-backed AMI use the instance's instance store as the root device (for example, C:\). An instance launched from an AMI backed by Amazon EBS uses an Amazon EBS volume as its root device. The **Root Device** column on the console indicates whether the AMI is backed by an instance store (instance-store) or Amazon EBS (ebs). There are significant differences between the two types of AMIs. It is essential to consider those differences when you choose an AMI to launch your instance. Refer to the following section for a quick summary of the differences.

After locating your desired AMI, record its AMI ID. You can use the AMI ID to launch and then connect to your instance. For more information on launching and using your Windows instance, go to [Launching and Using Instances](#). For information on connecting to your Windows instance, go to [Connecting to Windows Instances](#).

Root Device Storage on Windows AMIs

The root device of an *instance store-backed AMI* is initially stored in Amazon S3. When an instance is launched using an instance store-backed AMI, an image of the root device is copied from Amazon S3 to the root partition of the instance's instance store. This image is then used to boot the instance.

The root device of an [Amazon EBS-backed AMI](#) is an Amazon EBS snapshot. When an instance is launched using the Amazon EBS-backed AMI, a root EBS volume is created from the EBS snapshot and is attached to the instance. The root volume is then used to boot the instance.

There are other important differences between the two types of AMIs. The following table gives a quick summary of the differences.

Characteristic	Amazon EBS-Backed	Amazon Instance Store-Backed
Boot Time	Usually less than 1 minute	Usually less than 5 minutes
Size Limit	1 TiB	10 GiB
Root Device Location	Amazon EBS volume	Instance store
Data Persistence	Data persists on instance failure and can persist on instance termination	Data persists for the life of the instance; nonroot devices can use Amazon EBS
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	Instance usage, Amazon EBS volume usage, and Amazon EBS snapshot charges for AMI storage	Instance usage and Amazon S3 charges for AMI storage
Stopped State	Can be placed in stopped state where instance is not running, but the instance is persisted in Amazon EBS	Cannot be in stopped state; instances are running or not

Creating Your Own Windows AMI

When you are connected to your Amazon Windows EC2 instance, you can use it just like you use a hardware-based Windows Server instance, but with cloud functionality. There are several ways you can use your Windows instance:

- Use the instance as is for a specific task and duration, and stop or terminate the instance when your task is done.
- Customize your instance by installing software, applications, and additional storage to use the instance for a specific task and duration. For example, you can use an Amazon Windows-2008R2-SP1-MultiLang-SQLStandard AMI as a base, install Visual Studio Team Foundation Server, and attach Amazon EBS volumes for additional storage.



Note

An Amazon Windows EC2 instance can be rebooted after installing software and applications. Rebooting works the same way for both *instance store-backed* and *Amazon EBS-backed instances*.

- You can create your own AMI from your customized instance. This new customized AMI can then be used as a base to launch multiple instances.

For information on launching, connecting, and using your Windows instance, see [Using Instances](#).

Before you create your own AMI, you can configure your base customized instance. The new configuration will be applied to all the instances that will be launched from the new AMI. Your Amazon Windows instance comes with a configuration tool, EC2Config Service. You can use this tool to configure your instance. For information on using the EC2 Config Service, see [Using EC2Config Service \(p. 18\)](#)

The root storage device you select for the AMI determines the process you follow to create the AMI. The AMI will be either an Amazon EBS-backed AMI or an Amazon EC2 instance store-backed AMI. There are significant differences between Amazon EBS-backed and Amazon EC2 instance store-backed AMIs, including AMI size limits and storage and persistence of data. For information on the differences between these choices, see [Root Device Storage on Windows AMIs \(p. 17\)](#).

For detailed instructions on creating an Amazon EBS-backed Windows AMI, see [Creating an Amazon EBS-Backed Windows AMI \(p. 23\)](#). For detailed instructions on creating an instance store-backed Windows AMI, see [Creating an Instance Store-Backed Windows AMI \(p. 25\)](#).

Using EC2Config Service

EC2Config service can perform various functions to prepare an instance when it first boots up. Each of these functions can be enabled or disabled from the settings before creating a new AMI. The new settings are applied to the instance that will be launched from the new AMI.

EC2Config settings can be `enabled` or `disabled` using either the EC2Config Service Properties user interface tool or by directly editing the XML files. Some advanced configuration settings are currently not available on the user interface tool. For advanced modifications use the XML files. The following procedures describe the user interface tool and the XML files.

We Sysprep all the Amazon base Windows AMIs before registering and making them available to the public. So, when you launch an Amazon Windows AMI, it has to go through the initial Sysprep cycle. Sysprep is a Microsoft tool that prepares an AMI for multiple launches. However, after you launch your instance and the instance configures itself, Sysprep should not run unless you manually invoke it.

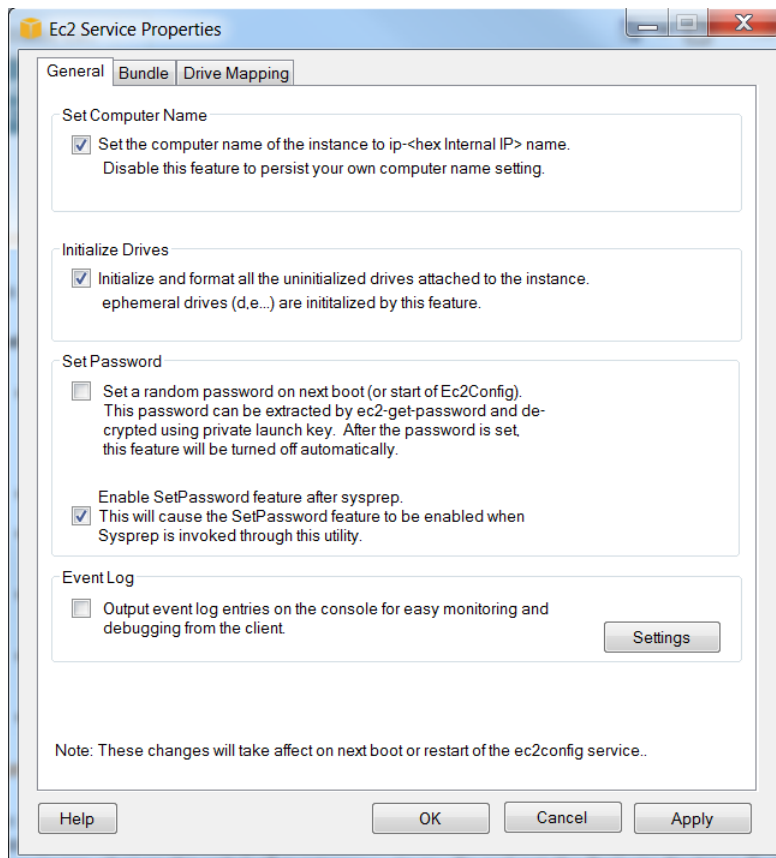
You can manually invoke the Sysprep tool to prepare your instance to create a new Windows AMI. The process of customizing and preparing an instance for creating a new AMI is also called *bundling*. The following procedure describes steps to bundle your instance using the EC2 Service Properties user interface.

Using the EC2Config Service User Interface Tool

Topics

To change the EC2Config settings on your Windows instance using the EC2Config tool

1. Launch and connect to your Windows instance.
2. Go to `C:\Program Files\Amazon\EC2ConfigService` and double-click the **Ec2ConfigServiceSettings** application.
3. Your Windows instance displays the **Ec2 Service Properties** dialog box.



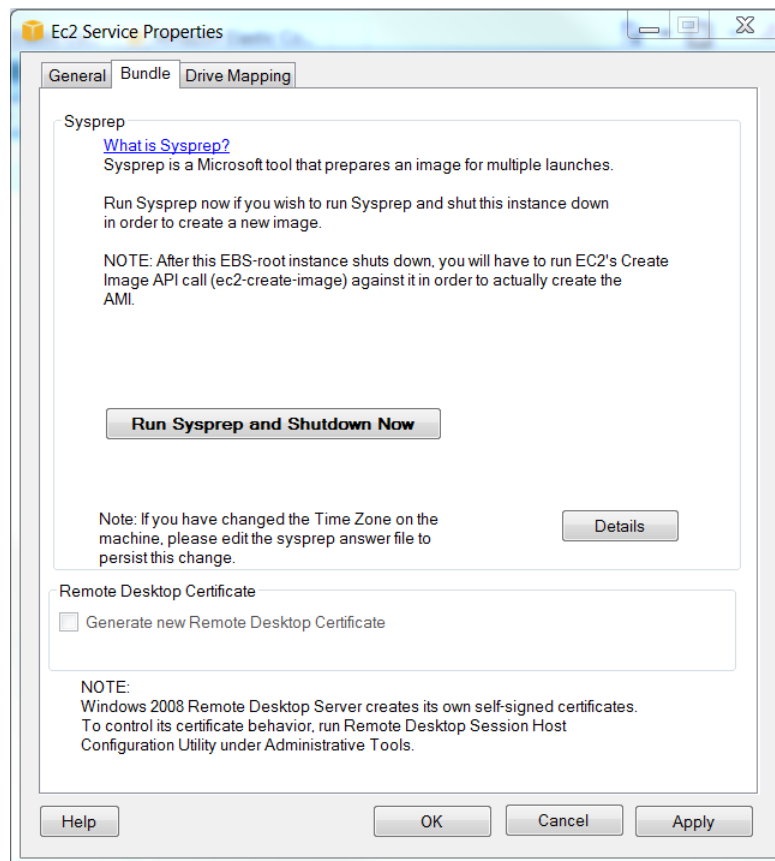
4. Use the **Ec2 Service Properties** dialog box to enable or disable your settings. In the **General Properties** tab you can adjust the following settings:
 - a. **Set Computer Name**—Enabled by default, sets the hostname of the instance to a unique name based on the IP address of the instance and reboots once after booting. If you want to set your own hostname, or prevent your existing host name from being modified, you must disable this setting.
 - b. **Initialize Drives**—Initializes and formats all uninitialized instance stores attached to the instance during startup. When an instance is launched, all instance stores that come with the instance

Amazon Elastic Compute Cloud Microsoft Windows Guide Using EC2Config Service

are uninitialized. Enabled by default, this feature initializes and mounts the instance stores as drives D:/, E:/, etc. For more information on instance stores that come with Windows instances, see [Amazon EC2 Instance Storage](#).

- c. **Set Password**—Sets a random password on the instance every time you launch an instance, encrypts it with the user launch key, and outputs the encrypted password to the console. This feature is disabled by default after the first launch so that any further reboots or restarts of this instance do not change the password set by the user. Click the empty check box to enable this setting to continue generating random passwords every time you launch this instance.
- d. **Enable SetPassword feature after sysprep**—Enabled by default, sets a random password after you have used sysprep feature to create an AMI from this instance.
- e. **Event Log**—Click the empty check box to enable this setting to put eventlog entries on the console during boot for easy monitoring and debugging. Click **Settings>** in the **EventLog** box to specify filters for the log entries that will be sent to the console output. By default, the three most recent error entries from the System event log are sent to the console.

5. Click the **Bundle** tab to open the Sysprep page.

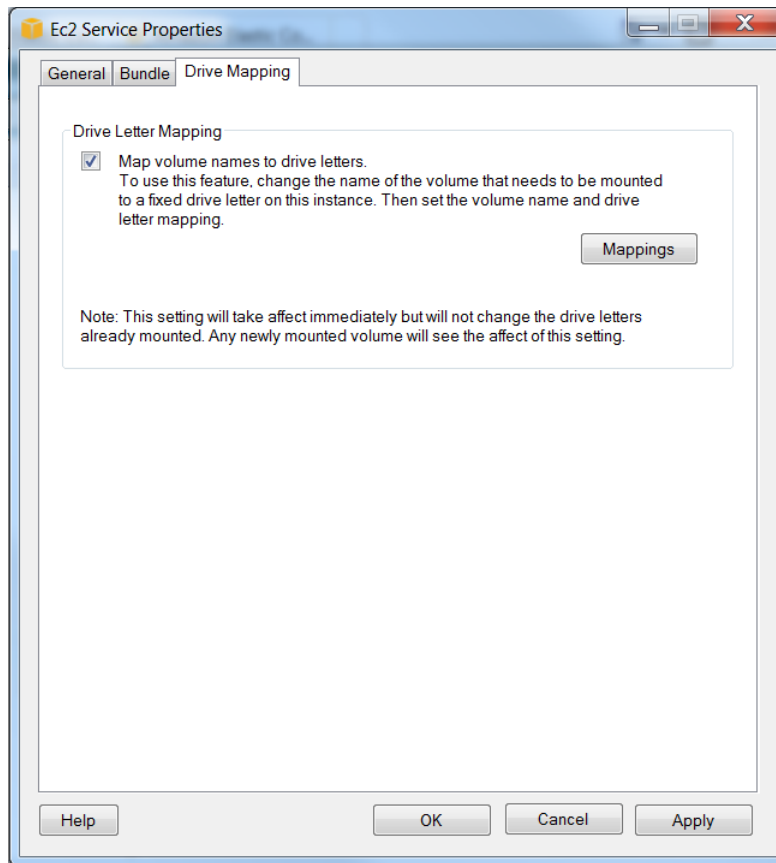


6. Click **Run Sysprep and Shutdown Now** to prepare your instance for creating a new AMI, click **Apply** and then click **OK**.

By default, when an Amazon EBS volume is attached to an instance, it may show up as any drive letter on the instance. You can specify the drive letters of the mounted volumes by mapping the name of a volume to a drive letter.

7. Click the **Drive Mapping** tab to specify your drive mapping.

The **Drive Letter Mapping** setting is enabled by default. This means that the drives will be mapped to drive letters, but the system decides the mapping.



8. To specify your own mapping, click **Mappings**.
 - a. In the **DriveLetterSetting** dialog box, type in the name (disk label) of the volume.
 - b. Click the drop-down arrow in the **Drive Letter** box and select the drive letter for the volume.
 - c. Keep adding the volume names and the drive letters you want to map it with.
 - d. Click **OK** to close the **DriveLetterSetting** dialog box.

9. Click **OK** to close the **Ec2 Service Properties** dialog box.

If you have specified your drive letter mapping, the settings will take effect immediately on volumes you attach after following this procedure. This setting will not change the drive letters on already mounted volumes. However, the setting will fail if the drive letter is already in use. We recommend that you pick drive letters from the end of the alphabet (Z, Y, X, and so on) to avoid this problem.

Using EC2Config XML Files

Topics

By default, the EC2Config service is installed on all public Amazon AMIs. The binaries and additional tools needed to configure the new Windows AMI are contained in the

`%ProgramFiles%\Amazon\EC2ConfigService` directory on 32-bit instances and in the `%ProgramFiles(x86)%\Amazon\EC2ConfigService` directory on 64-bit instances. You can modify the following configuration and settings files located in the directory.

Config.xml File

- **Ec2SetPassword**—Generates a new random encrypted password every time you launch an instance. This feature is disabled by default after the first launch so that any further reboots or restarts of this instance do not change the password set by the user. Change this setting to `Enabled` to continue generating random passwords every time you launch an instance.

This setting is important, if you are planning on creating an AMI from your instance. Before you configure this setting you have to decide whether you want the instances launched from the customized AMI to have random passwords generated.

- **Ec2SetComputerName**—Enabled by default, sets the hostname of the instance to a unique name based on the IP address of the instance and reboots once after booting. If you want 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 uninitialized instance stores attached to the instance during startup. When an instance is launched, all instance stores that come with the instance are uninitialized. `Enabled` by default, this feature initializes and mounts the instance stores as drives D:/, E:/, etc. For more information on the instance stores that come with Windows instances, go to [Amazon EC2 Instance Storage](#).
- **Ec2EventLog**—Puts event log entries in the console based on the configuration of the `eventlogconfig` file. By default, the three most recent error entries from the system event log are sent to the console. To specify the event log entries to output to the console, edit the **EventLogConfig.XML** file located in the **Settings** directory.

For information on the settings in this file, go to the [Microsoft MSDN website](#).

- **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 instances since Windows Server 2008 is able to generate its own certificates.
- **Ec2OutputRDPcert**—This setting is `Enabled` by default and copies the Remote Desktop certificate information to the console, so the user can verify it against the thumbprint.
- **Ec2SetDriveLetter**—Sets the drive letters of the mounted volumes based on the user defined settings. By default, when an Amazon EBS volume is attached to an instance, it may show up as any drive letter on the instance. To specify your drive letter mappings, edit the **DriveLetterConfig.XML** file located in the **Settings** directory.
- **Ec2WindowsActivate**—Enabling this setting causes Windows Server 2008 to attempt activation by searching through the DNS Suffix List for appropriate KMS Server entries. In Windows Server 2008 (not the later Windows Server R2 version), the plug-in performs this search manually. When the appropriate KMS server entries are found, the plug-in sets your activation server to the first server to respond to the request successfully. In Windows Server 2008 R2, the auto discovery method built into Windows Server is able to search the suffix list automatically.

To modify the settings for the KMS servers, edit the `ActivationSettings.XML` file located in the `Settings` directory.

- **SetDnsSuffixList**—Adds entries to your DNS suffix search list to facilitate DNS lookups.



Note

This functionality is key to Windows Server 2008 activation. See the preceding list item for more information.

This plug-in has the ability to make region-specific decisions about the suffix list based on the Availability Zone the instance has been launched in. The default settings have been configured to discover the

KMS server in each region. To see the default settings or add settings specific to your requirements, open and edit the **DnsSuffixSettings.XML** file located in the **Settings** directory.

The `BundleConfig.xml` file controls how the EC2Config service prepares an instance for bundling. This includes configuring sysprep on the system, changing the state of the Ec2ConfigureRDP plug-in, and shutting down the instance for bundling.

BundleConfig.XML

- **AutoSysprep**—Change the value to `Yes` if you want to use sysprep.
- **SetRDPCertificate**—Sets a self-signed certificate to the Remote Desktop server running on a Windows 2003 instance. This allows you to securely RDP into the instances. Change the value of this setting to `Yes` if you want the new instances to have the certificate.



Note

This setting is not used in Windows Server 2008, since Windows Server 2008 is able to generate its own certificates.

- **SetPasswordAfterSysprep**—Sets a random password on the newly launched instances, encrypts it with the user launch key, and outputs the encrypted password on the console. Change the value of this setting to `No` if you do not want the new instances to set a random encrypted password.

Creating an Amazon EBS-Backed Windows AMI

The process for creating an Amazon EBS-backed AMI is simple. The following diagram and the task lists describe the process.



Task List for Creating an Amazon EBS-backed AMI

1. Start by launching an instance of an Amazon EBS-backed AMI that is similar to the AMI you want to create. For example, you might take a public AMI that uses the operating system you want to use for your AMI.
The instance must be from an Amazon EBS-backed AMI; you can't start with an instance of an Amazon EC2 instance store-backed AMI.
2. When the instance is running, customize it as you want. For example, you could attach additional Amazon EBS volumes, or load applications or data on to the instance.



Important

If you customize your instance with additional instance stores, the new AMI contains the mapping information for those instance stores. When you then launch an instance from your new AMI, the instance automatically launches with the additional instance stores initialized and mounted. However, all the attached instance stores are new. They will not contain any data from the instance store of the original AMI.



Important

If you customize your instance with EBS volumes in addition to the root device, the new AMI contains block device mapping information for those volumes. When you then launch an instance from your new AMI, the instance automatically launches with the additional volumes. The data on those volumes will persist.

3. When an instance is launched for the first time, it goes through a cycle of configuration and Sysprep. After the instance boots, some of the configuration settings are either `Enabled` or `Disabled` to persist the settings through instance reboots. Use EC2Config Services tool or EC2Config XML files to change the configuration settings for your new AMI.
4. When the instance is set up the way you want it, create an AMI from that instance.

Amazon EC2 powers down the instance, takes images of any volumes that were attached, creates and registers the AMI, and then reboots the instance. It takes several minutes for the entire process to complete.

You can use the command line, the API, or the AWS Management Console to create your Windows AMIs. For background information about the interfaces and tools, see [Available Interfaces \(p. 9\)](#).

Using the AWS Management Console

For instructions on using the AWS Management Console, go to [Creating an Image from a Running Instance](#).

Using Command Line Tools

To create an Amazon EBS-backed AMI

1. Enter the following command to create an image:

```
C:\> ec2-create-image -n your_image_name instance_id
```

For example:

```
C:\> ec2-create-image -n "My AMI" i-eb977f82
```

Amazon EC2 creates an image and returns an AMI ID.

```
IMAGE ami-8675309
```

2. If you want to check whether the AMI is ready, enter the following command:

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

Amazon EC2 returns information about the AMI.

Using the Query API

To create an Amazon EBS-backed AMI

- Construct the following query request to create an image:

```
https://ec2.amazonaws.com/  
?Action=CreateImage  
&InstanceId=instance_id  
&Name=My_Ami  
&...auth parameters...
```

In the following example response, Amazon EC2 creates the image and returns its AMI ID.

```
<CreateImageResponse xmlns="http://ec2.amazonaws.com/doc/2011-07-15/">  
  <imageId>ami-8675309</imageId>  
</CreateImageResponse>
```

AMI creation can take time. You can check whether the AMI is ready by using [DescribeImages](#).

Creating an Instance Store-Backed Windows AMI

The process for creating an *instance store-backed* AMI is different from the process of creating an *Amazon EBS-backed* AMI. All Amazon EC2 instance store-backed AMIs are loaded from Amazon S3 storage. When you create the AMI, you must upload it to an existing account on 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. We refer to this process of creating an AMI as *Bundling*.



Caution

Instance store drives (e.g., the D: drive) are not included in the bundled AMI. Instance store drives and their data are deleted when the instance is terminated. You must store any data that you want to use with the new AMI on the root drive or an Amazon EBS volume. For more information about Amazon EC2 storage options, go to [Using Storage](#).

You cannot launch the new AMI until the bundling is complete and you have registered the bundled image. The bundling process can take time and you can monitor the task by using the `ec2-describe-bundle-tasks` command. While bundling is in progress, the task moves through a succession of states, including "waiting-for-shutdown," "storing," and "complete" states. The output during the process looks like this:

```
BUNDLE bun-1509ed7c i-cb2a81a0 mybucket myimage 2010-03-19T08:22:48+0000 2010-  
03-19T08:23:50+0000 bundling 12%
```

When bundling is complete, the status changes to "complete."

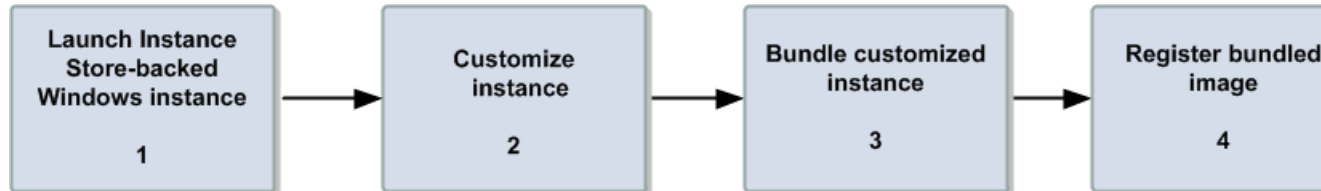
You must register your bundled image with Amazon EC2, so Amazon EC2 can locate it and run instances based on it. You don't have to register the bundled image immediately after the bundle task completes. You can still register the bundled image even if the bundle task no longer appears in your list of completed bundle tasks (each task remains on the list for only a limited time).



Note

If you make any changes to the source image stored in Amazon S3, you must reregister the image.

The following diagram shows the general tasks in creating Amazon EC2 instance store-backed Windows AMIs.



Tasks for Creating an Instance Store-Backed Windows AMI

1. Start by launching a Windows instance that is similar to the AMI you want to create and connect to the instance. For information on connecting to a Amazon EC2 Windows instance, go to [Connecting to Windows Instances](#).
2. When you are connected to your instance, customize it as you want. For example, you could attach Amazon EBS volumes, or load applications or data on to the instance.



Important

It is very important to remember the following behavior of the *instance store-backed* AMI creation process on the storage drives:

- The system drive (C:) is automatically attached when the new instances are launched. The data on the C: drive is the only one that will persist when the instance is bundled.
 - Other instance store drives (for example, D:) are temporary drives, and should be strictly used for short-term storage, since they will not persist when instance is bundled.
 - You can add Amazon EBS volumes. The EBS volumes are stored within Amazon S3 buckets and therefore will remain intact when the instance is bundled. We recommend that you store all the data that you need to persist within the Amazon EBS volumes.
3. When an instance is launched for the first time, it goes through a cycle of configuration and Sysprep. After the instance boots, some of the configuration settings are either *Enabled* or *Disabled* to persist the settings through instance reboots. Before bundling an instance, you can configure the instance using the EC2Config Service. Use EC2Config Services tool or EC2Config XML files to change the configuration settings for your new AMI. For more information, see [Using EC2Config Service \(p. 18\)](#).
 4. Bundle and then register the AMI.
 5. After you've prepared your instance, the bundling process performs the following tasks, which are listed in the order that they usually take place:
 - Excludes any instance store drives (i.e., the D: drive on your instance is not included in the bundled AMI)
 - Compresses the image to minimize bandwidth usage and storage requirements
 - Encrypts and signs the compressed image to ensure confidentiality and authenticates the image against its creator
 - Splits the encrypted image into manageable parts for upload
 - Runs `sysprep` to strip out computer-specific information (e.g., the MAC address and computer name) to prepare the Windows image for virtualization

- Creates a manifest file that contains a list of the image parts with their checksums
- Puts all the parts of the AMI into an Amazon S3 bucket you specify

How to Bundle Amazon EC2 Instance Store-Backed Windows AMIs

You can use the AWS Management Console, command line, or API to bundle Amazon EC2 instance store-backed Windows AMIs.

Using the AWS Management Console

Bundling and registering Amazon EC2 instance store-backed Windows images using the console is easy.

To bundle an Amazon EC2 instance store-backed AMIs

1. Log in to the [AWS Management Console](#) and click the box next to the drop-down arrow to open a list of services. Click Amazon Elastic Compute Cloud and then select the Windows instance you want to use, and prepare the instance to meet your specifications.

For information about running an instance, go to [Running an Instance](#).

2. Right-click the instance you customized and select **Bundle Instance (S3 AMI)**.

The **Bundle Instance** dialog box opens. It shows the ID of the instance you want to bundle.

3. Provide your **Amazon S3 Key Name** and the **Amazon S3 Bucket Name** where you want the new AMI to be stored, and then click **Bundle**.

You should see the **Bundle Instance** message box informing you that you successfully made the bundling request. The message box also provides the **Bundle Task ID**.

Click **View Bundling Tasks** to see the status of the task. Click **Close** to close the message box.



Note

The **Bundle Tasks** status can show **waiting-for-shutdown** when Amazon EC2 is bundling an Amazon EC2 instance store-backed instance. Amazon EC2 shuts down the instance, bundles it, and puts the new bundle into Amazon S3.

4. Navigate to the list of your AMIs when the bundling task is complete, right-click the newly-bundled AMI, and then select **Register New AMI**.

The **Register Image** dialog box opens. Provide the **AMI Manifest Path** and click **Register**.

Using the Command Line Tools

To bundle Amazon EC2 instance store-backed AMIs

1. Log in to the Windows instance and modify it to meet your requirements.



Note

We recommend that you change the password of the AMI. If you use the Amazon EC2-provided password, write it down so you can access instances launched from this AMI. You cannot get the password for new instances using the `ec2-get-password` command.

**Amazon Elastic Compute Cloud Microsoft Windows
Guide
Creating an Instance Store-Backed Windows AMI**

2. If you want to reduce your startup time, delete any temporary files on your instance using the Disk Cleanup tool, defragment your system using Disk Defragmenter, and zero out free space using `sdelete -c C:\`.
You can download the `sdelete` utility from the [sdelete Download Page](#) or the [Microsoft Web Site](#).
3. Enter the following command to bundle the instance into Amazon S3 on your local system where you have installed the API tools (do not enter this command on the instance you are bundling):

```
C:\> ec2-bundle-instance <instance_id> -b <bucket_name> -p <bundle_name> -o  
<access_key_id> -w <secret_access_key>
```

The `<instance_id>` is the name of the instance; `<bucket_name>` is the name of the bucket in which to store the AMI; and `<bundle_name>` is the common name for the files to store in Amazon S3.



Note

To perform this task, you need your AWS Access Key ID (`access_key_id`) and AWS Secret Access Key (`secret_access_key`). For more information, go to [Amazon EC2 Credentials](#).

The `ec2-bundle-instance` utility uploads the bundled AMI to a specified bucket. If you have used Amazon S3 before, you can use any of your existing buckets or just give `ec2-bundle-instance` any name that makes sense to you. If the specified bucket does not exist, the command creates it. If the specified bucket belongs to another AWS account, `ec2-bundle-instance` fails, and you have to specify a different name.

The following is an example of a fully specified `ec2-bundle-instance` command.

```
C:\> ec2-bundle-instance -31c2425a -b mybucket -p myimage -o AKIADQKE4SARGYLE  
-w eW91dHVizS5jb20vd2F0Y2g/dj1SU3NKMTlzeTNKSQ==  
BUNDLE bun-e3a4418a i-31c2425a mybucket myimage 2010-03-  
19T08:22:48+0000 2010-03-19T08:22:48+0000 pending
```

Amazon EC2 shuts down the instance, saves it as an AMI, and restarts it.

4. Enter the following command to register the image:

```
C:\> ec2-register <your-s3-bucket>/image.manifest.xml -n image_name  
IMAGE ami-2bb65342
```

Amazon EC2 returns an AMI identifier, the value next to the `IMAGE` tag (`ami-2bb65342` in the example) that you can use to run instances.

Using Query API

The following procedure steps you through how to bundle an Amazon EC2 instance store-backed AMI. This process mirrors the steps you would use with the command line tools.

To bundle Amazon EC2 instance store-backed AMIs

1. Log in to the Windows instance and modify it to meet your requirements.
2. Construct the following request to bundle the instance into Amazon S3 on your local system where you have installed the API tools (do not construct the request on the instance you are bundling):

Amazon Elastic Compute Cloud Microsoft Windows Guide Shared Windows AMIs

```
https://ec2.amazonaws.com/  
?Action=BundleInstance  
&InstanceId=-i-e468cd8d  
&Storage.S3.AWSAccessKeyId=10QMXFEV71ZS32XQFTR2  
&Storage.S3.Bucket=my-bucket  
&Storage.S3.Prefix=winami  
&Storage.S3.UploadPolicy=eyJleHBpcmF0aW9uIjogIjIwMDgtMDgtMzBUMDg6NDk6MDlaIiw  
wiY29uZGl0aW9ucyI6IjFt7ImJlY2tldCI6ICJteS1idWNrZXQifSxbInN0YXJ0cy13aXRoIiw  
gIiRrZXkiLCAibXktbmV3LWltYWdlIl0seyJhY2wiOiAiZWMyLWJlbnRsZS1yZWfkIn1dfQ%3D%3D  
&Storage.S3.UploadPolicySignature=fh5tyyyQD8W4COEthj3nLGNtJMU%3D  
&AuthParams
```



Note

To perform this task, you need your AWS Access Key ID (<aws-access-key-id>) and AWS Secret Access Key (<aws-secret-access-key>). For more information, go to [Amazon EC2 Credentials](#).

For information about the `BundleInstance` command, see [BundleInstance](#) in the *Amazon Elastic Compute Cloud API Reference*.

3. Construct the following command to register the image:

```
https://ec2.amazonaws.com/  
?Action=RegisterImage  
&ImageLocation=full-path-to-amazon-manifest  
&AuthParams
```

Amazon EC2 returns an AMI identifier that you can use to run instances.

Shared Windows AMIs

Shared Windows AMIs are the Windows AMIs that developers build and make available for other AWS developers to use. You can either use an available shared AMI or create your own AMI for sharing. Creating safe, secure, usable Windows AMIs for public consumption is a fairly straightforward process.

Creating Windows AMIs for Sharing

Following these guidelines produces a better user experience, makes your users' *instances* less vulnerable to security issues, and helps protect you.

To create a Windows AMI for sharing, follow these guidelines:

1. Follow the instructions to launch and connect to a Windows instance.
2. Customize the instance by installing the software and applications you want to share. Do the following to make your AMI safe and secure for sharing:
 - Always delete the shell history before bundling. The shell history may contain sensitive information.
 - If you have saved your instance credentials, such as your key pair, remove them or move them to a location that is not going to be included in the AMI.
 - Ensure that the Administrator password, and passwords on any other accounts, is set to an appropriate value for sharing. These passwords will be available for anyone who launches your shared AMI.

- Remove any saved passwords.
 - Make sure to test your AMI before you release to the public.
3. Run sysprep to prepare the instance and enable the new password generation on new instance launch. The instance will shut down.
 4. Create an image of the instance.

Sharing AMIs

Amazon EC2 enables you to share your AMIs with other AWS accounts. This section describes how to share AMIs using the Amazon EC2 command line tools.



Note

Before proceeding, make sure to read the security guidelines for sharing AMIs in the [Creating Windows AMIs for Sharing \(p. 29\)](#).

AMIs have a `launchPermission` property that controls which AWS accounts, besides the owner's, are allowed to launch instances of that AMI. By modifying an AMI's `launchPermission` property, you can allow all AWS accounts to launch the AMI (i.e., make the AMI public) or only allow a few specific accounts to launch the AMI.

The `launchPermission` attribute is a list of accounts and *launch groups*. *Launch permissions* can be granted by adding or removing items from the list. Explicit launch permissions for accounts are granted or revoked by adding or removing AWS account IDs. The only launch group currently supported is the `all` group, which makes the AMI public. The rest of this section refers to launch groups simply as groups. Launch groups are not the same as security groups and the two should not be confused. An AMI can have both public and explicit launch permissions.



Note

You are not billed when your AMI is launched by other AWS accounts. The accounts launching the AMI are billed.

Making an AMI Public

To make an AMI public

- Add the `all` group to the AMI's `launchPermission`.

```
C:\> ec2-modify-image-attribute <ami_id> --launch-permission -a all
```

The `<ami_id>` parameter is the ID of the AMI.

This example makes the `ami-2bb65342` AMI public.

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

To check the launch permissions of an AMI

- Enter the following command, where `<ami_id>` is the ID of the AMI.

```
C:\> ec2-describe-image-attribute <ami_id> -l
```

This example displays the launch permissions of the ami-2bb65342 AMI.

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

To make an AMI private again

- Remove the `all` group from its launch permissions, where `<ami_id>` is the ID of the AMI.

```
C:\> ec2-modify-image-attribute <ami_id> -l -r all
```

This will not affect any explicit launch permissions for the AMI or any running instances of the AMI.

This example removes the `all` group from the permissions of the ami-2bb65342 AMI, making it private.

```
C:\> ec2-modify-image-attribute ami-2bb65342 -l -r all
launchPermission      ami-2bb65342      REMOVE  group    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 is the account ID.

To grant explicit launch permissions

- Enter the following command:

```
C:\> ec2-modify-image-attribute <ami_id> -l -a <user_id>
```

The `<ami_id>` is the ID of the AMI and `<user_id>` is the account ID, without hyphens.

The following example grants launch permissions to the AWS account with ID 999988887777 for the ami-2bb65342 AMI:

```
C:\> ec2-modify-image-attribute ami-2bb65342 -l -a 999988887777
launchPermission      ami-2bb65342      ADD     userId   999988887777
```

To remove launch permissions for an account

- Enter the following command:

Amazon Elastic Compute Cloud Microsoft Windows Guide Creating Windows AMIs for Sharing

```
C:\> ec2-modify-image-attribute <ami_id> -l -r <user_id>
```

The `<ami_id>` is the ID of the AMI and `<user_id>` is the account ID, without hyphens.

The following example removes launch permissions from the AWS account with ID 999988887777 for the ami-2bb65342 AMI:

```
C:\> ec2-modify-image-attribute ami-2bb65342 -l -r 999988887777  
launchPermission      ami-2bb65342      REMOVE      userId      999988887777
```

To remove all launch permissions

- Enter the following command to remove all public and explicit launch permissions:

```
C:\> ec2-reset-image-attribute <ami_id> -l
```

The `<ami_id>` is the ID of the AMI.

The following example removes all public and explicit launch permissions from the ami-2bb65342 AMI:

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



Note

The AMI owner always has rights to the AMI and is unaffected by this command.

Publishing Shared AMIs

After you create a shared AMI, you can publish information about it in the [Amazon EC2 Resource Center](#).

To publish your AMI

1. Post your AMI in the Public AMIs folder of the [Amazon Web Services Resource Center](#), and include the following information:
 - AMI ID
 - AMI name (for Amazon EBS-backed AMIs) or AMI manifest (for Amazon EC2 instance store-backed AMIs)
 - Publisher
 - Publisher URL
 - OS / Distribution
 - Key feature
 - Description
 - Daemons / Services
 - Release Notes
2. If you want to, you can paste the following information into the document. You must be in HTML edit mode.

The following examples show how to use a flag to filter the results.

- The following command displays a list of all public AMIs. The `-x all` flag shows AMIs executable by all AWS accounts (i.e., AMIs with public launch permissions). This includes AMIs you own with public launch permissions.

```
C:\> ec2dim -x all
```

- The following command displays a list of AMIs for which you have explicit *launch permissions*. AMIs that you own are excluded from the list.

```
C:\> ec2dim -x self
```

- The following command displays a list of AMIs owned by Amazon.

```
C:\> ec2dim -o amazon
```

- The following command displays a list of AMIs owned by a particular AWS account.

```
C:\> ec2dim -o <target_uid>
```

The `<target_uid>` is the account ID that owns the AMIs you're looking for.

For more information about the flags and how to use flags to filter the results, go to [ec2-describe-images](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

Safe Use of Shared AMIs

You launch AMIs at your own risk. Amazon cannot vouch for the integrity or security of AMIs shared by other 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.

Ideally, you should get the AMI ID from a trusted source (such as a website or another EC2 user that you trust). If you do not know the source of an AMI, we recommend that you search the AWS forums for comments on the AMI before launching it. Conversely, if you have questions or observations about a shared AMI, feel free to use the [AWS forums](#) to ask or comment.

Amazon's public images have an aliased owner and display `amazon` in the `userId` field. This allows you to find Amazon's public images easily.



Note

Users cannot alias an AMI's owner.

For information on launching, connecting, and using the Windows instances, see [Using Instances](#).

Paid Windows AMIs

This section describes how to discover paid AMIs, launch paid AMIs, and launch instances with a support product code. Paid AMIs are AMIs you can purchase from other developers.

Amazon EC2 integrates with Amazon DevPay, allowing developers to charge other EC2 users for the use of their AMIs or to provide support for instances. To learn more about Amazon DevPay go to the [Amazon DevPay Developer Guide](#).



Note

All paid AMIs are backed by Amazon instance store.

Find Paid AMIs

There are several ways you can determine what paid AMIs are available for you to purchase. You can look for information about them on the Amazon EC2 resource center and forums. Alternatively, a developer might give you information about a paid AMI directly.

You can also tell if an AMI is a paid AMI by describing the image with the `ec2-describe-images` command. This command lists the product code associated with an AMI (see the following example). If the AMI is a paid AMI, it has a product code. Otherwise, it does not. You can then go to the Amazon EC2 resource center and forums, which might have more information about the paid AMI and where you can sign up to use it.



Note

You must sign up for a paid AMI before you can launch it.

To check if an AMI is paid

- Enter the following command:

```
C:\> ec2-describe-images <ami_id>
```

The `<ami_id>` is the AMI ID.

The command returns numerous fields that describe the AMI. If a product code (e.g., `D6F6052A`) is present in the output, the AMI is a paid AMI.

This example shows an `ec2-describe-images` call describing a paid AMI. The product code is `ACD42B6F`.

```
C:\> ec2-describe-images ami-a5bf59cc
IMAGE    ami-a5bf59cc    cloudmin-2.6-paid/image.manifest.xml    541491349868
         available public    ACD42B6F            i386    machine
         instance-store
```

Purchase a Paid AMI

You must sign up for (purchase) the paid AMI before you can launch it.

Typically a seller of a paid AMI presents you with information about the AMI, its price, and a link where you can buy it. When you click the link, you're first asked to log in with an Amazon.com login, and then you are taken to a page where you see the paid AMI's price and you confirm you want to purchase the AMI.



Important

You don't get the discount from Amazon EC2 Reserved Instances with paid AMIs. That is, if you purchase Reserved Instances, you don't get the lower price associated with them when you launch a paid AMI. You always pay the price that the seller of the paid AMI specified. For more information about Reserved Instances, go to [On-Demand and Reserved Instances](#).

Launch a Paid AMIs

This section describes how to launch paid AMIs and launch instances with a support product code.

After you purchase a paid AMI, you can launch instances of it. Launching a paid AMI is the same as launching any other AMI. No additional parameters are required. The instance will be charged according to the rates set by the owner of the AMI.

To launch a paid AMI

- Enter the following command:

```
C:\> ec2-run-instances <ami_id>
```

The `<ami_id>` is the AMI ID.

This example shows the command used to launch the ami-2bb65342 AMI.

```
C:\> ec2-run-instances ami-2bb65342
RESERVATION r-a034c7c9 999988887777 default
INSTANCE i-31a7425a ami-2bb65342 pending 0 m1.small 2010-03-19T13:59:03+0000
us-east-1a aki-94c527fd ari-96c527ff monitoring-disabled ebs
```



Note

The owner of a paid AMI will be able to confirm if a particular instance was launched using their paid AMI.

Using Paid Support

The paid AMI feature also allows developers to offer support for software (or derived AMIs). Developers can create support products that you can sign up to use. With this model, the developer provides you with a product. During sign-up for the product, the developer gives you a product code for that product, which you must then associate with your own AMI. This allows 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 developer's terms for the product.



Important

If you've purchased Amazon EC2 Reserved Instances, you can't use them with supported AMIs. That is, if you associate a product code with one of your AMIs, you don't get the lower price associated with your Reserved Instances when you launch that AMI. You always pay the price that the seller of the support product specified. For more information about Reserved Instances, go to [On-Demand and Reserved Instances](#).

To associate the product code with your AMI

- Enter the `ec2-modify-image-attribute` command:

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

The `<ami_id>` is the AMI ID and `<product_code>` is the product code.



Important

Once set, the product code attribute cannot be changed or removed.

To launch a paid AMI, no additional parameters are required for `ec2-run-instances`. The instance is charged according to the rates set by the AMI owner.

The following command launches the `ami-2bb65342` paid AMI.

```
C:\> ec2-run-instances ami-2bb65342
RESERVATION r-a034c7c9 999988887777 default
INSTANCE i-31a7425a ami-2bb65342 pending 0 m1.small 2010-03-19T13:59:03+0000
us-east-1a aki-94c527fd ari-96c527ff monitoring-disabled ebs
```

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 the paid or supported AMIs during the month. This bill is separate from your regular Amazon EC2 bill.

For information on the usage information for your paid and supported AMIs, go to [amazonpayments](#)) sign in page.

Installing the Amazon EC2 Command Line Tools on Windows

This section describes how to install the Amazon EC2 command line tools, a set of tools that you can run from the Windows command line that closely mimics the Amazon EC2 API functions.

Process for Installing the Command Line API Tools

Task 1: Download the Command Line Tool (p. 38)
Task 2: Set the JAVA_HOME Environment Variable (p. 38)
Task 3: Set the EC2_HOME Environment Variable (p. 39)
Task 4: Set the EC2_PRIVATE_KEY and EC2_CERT Environment Variables (p. 40)
Task 5: Set the Region (p. 42)

Task 1: Download the Command Line Tool

The command line tool is available as a ZIP file on the [Amazon EC2 Resource Center](#). The tool is written in Java and includes shell scripts for both Windows and Linux/UNIX/Mac OSX. The ZIP file is self-contained; no installation is required. You just download it and unzip it.

Some additional setup is required before you can use the tool. These steps are discussed next.

Task 2: Set the JAVA_HOME Environment Variable

The Amazon EC2 command line tool reads an environment variable (`JAVA_HOME`) on your computer to locate the Java runtime. The command line tool requires Java version 5 or later to run. Either a JRE or JDK installation is acceptable.

To set the `JAVA_HOME` environment variable

1. If you do not have Java 1.5 or later installed, download and install Java. To view and download JREs for a range of platforms, including Linux/UNIX and Windows, go to <http://java.oracle.com/>.
2. Set `JAVA_HOME` to the full path of the directory that contains a subdirectory named `bin` that in turn contains the Java executable. For example, if your Java executable is in `C:\jdk\bin`, set `JAVA_HOME` to `C:\jdk`.
 - a. On the machine from which you will connect to Amazon Web Services, click **Start**, right-click **Computer**, and then click **Properties**.



Note

These instructions are written for a Windows 7 client. The steps may vary depending on the version of Windows you are using.

- b. Click **Advanced system settings**.
- c. Click **Environment Variables**.
- d. Under **System variables**, click **New**.
- e. In **Variable name**, type `JAVA_HOME`.
- f. In **Variable value**, type the path to your Java version. For example, "`C:\Program Files (x86)\Java\jre7`"



Note

Don't include the `bin` directory in `JAVA_HOME`; that's a common mistake some users make. The command line tool won't work if you do.

- g. Click **OK**.
3. Add your Java directory to your path before other versions of Java.
 - a. In **System variables**, select **Path**, and then click **Edit**.
 - b. In **Variable values**, before any other versions of Java add `;%JAVA_HOME%\bin;`.
 4. Verify your `JAVA_HOME` setting with the command `%JAVA_HOME%\bin\java -version`.

```
C:\> %JAVA_HOME%\bin\java -version
java version "1.5.0_09"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_09-b03)
Java HotSpot(TM) Client VM (build 1.5.0_09-b03, mixed mode, sharing)
```

Task 3: Set the EC2_HOME Environment Variable

The command line tool depends on an environment variable (`EC2_HOME`) to locate supporting libraries. You'll need to set this environment variable before you can use the tool.

To set the EC2_HOME environment variable

1. Set EC2_HOME to the path of the directory into which you unzipped the command line tool. This directory is named `ec2-w.x.y.z` (`w`, `x`, `y`, and `z` are version/release numbers) and contains sub-directories named `bin` and `lib`.
 - a. On the machine from which you will connect to Amazon Web Services, click **Start**, right-click **Computer**, and then click **Properties**.



Note

These instructions are written for a Windows 7 client. The steps may vary depending on the version of Windows you are using.

- b. Click **Advanced system settings**.
 - c. Click **Environment Variables**.
 - d. Under **System variables**, click **New**.
 - e. In **Variable name**, type `EC2_HOME`.
 - f. In **Variable value**, type the path to the directory where you installed the command line tools. For example, "`C:\AWS\EC2\ec2-1.0.12.0`".
2. Add the tool's `bin` directory to your system `PATH`. The rest of this guide assumes that you've done this.

You can update your `PATH` as follows:

- a. In **System variables**, select **Path**, and then click **Edit**.
- b. In **Variable values**, add `;%EC2_HOME%\bin`.

Task 4: Set the EC2_PRIVATE_KEY and EC2_CERT Environment Variables

The command line tools need access to an X.509 certificate and a corresponding private key that are associated with your account. Amazon EC2 uses the certificate and private key to verify that the commands that you issue come from your account. You can create up to two pairs of X.509 certificates and private keys.

You can either specify your credentials with the `--private-key` and `--cert` parameters every time you issue a command or you can create environment variables that point to the credential files on your local system. If the environment variables are properly configured, you can omit the parameters when you issue a command.

You can use an existing X.509 certificate and private key, or you can create a new certificate and private key. The following procedure describes how to create a new X.509 certificate and private key and how to create environment variables that point to your credentials.

To set up security credentials for your command line tool

1. If you want to create a new certificate and private key, log in to the AWS [security credentials](#) website.

Amazon Elastic Compute Cloud Microsoft Windows Guide

Task 4: Set the EC2_PRIVATE_KEY and EC2_CERT Environment Variables

- a. Scroll down to the **Access Credentials** section and select the **X.509 Certificates** tab.
- b. Click **Create a new Certificate**.

The **X509 Certificate Created** page appears.



Note

The **Create a new Certificate** is available only if you have fewer than two existing X.509 certificates.

- c. Select **Download Private Key File** and save it to a convenient location on your computer.



Important

Store your Private Key file in a secure location. If you lose your Private Key file you will need to create a new certificate to use with your account. AWS does not store Private Key Information.

- d. Select **Download X.509 Certificate** and save it to a convenient location on your computer.
- e. Click **Close** to close the **X509 Certificate Created** page.

2. Set the `EC2_CERT` environment variable to the fully qualified path of an existing X.509 certificate or the one you just created.

- a. On the machine from which you will connect to Amazon Web Services, click **Start**, right-click **Computer**, and then click **Properties**.



Note

These instructions are written for a Windows 7 client. The steps may vary depending on the version of Windows you are using.

- b. Click **Advanced system settings**.
- c. Click **Environment Variables**.
- d. Under **System variables**, click **New**.
- e. In **Variable name**, type `EC2_CERT`.
- f. In **Variable value**, type the path to the directory where you saved your certificate. For example, `C:\aws\certs\cert\pk-B2QQ72ELI6RDHOYX6RK6ZPAW2SXTFUX.pem`.

3. Set the `EC2_PRIVATE_KEY` environment variable to the fully qualified path of the private key that is associated with the X.509 certificate that `EC2_CERT` now points to.

- a. On the machine from which you will connect to Amazon Web Services, click **Start**, right-click **Computer**, and then click **Properties**.



Note

These instructions are written for a Windows 7 client. The steps may vary depending on the version of Windows you are using.

- b. Click **Advanced system settings**.
- c. Click **Environment Variables**.
- d. Under **System variables**, click **New**.
- e. In **Variable name**, type `EC2_PRIVATE_KEY`.
- f. In **Variable value**, type the path to the directory where you saved your private key. For example, `C:\aws\certs\key\pk-pk-LJFUG4MXPQD6VAHJBBQFCQB7KHP4X3PX.pem`.

Task 5: Set the Region

By default, the Amazon EC2 tools use the Eastern United States Region (`us-east-1`) with the `ec2.us-east-1.amazonaws.com` service endpoint URL. If your instances are in a different region, you must specify the region where your instances reside. For example, if your instances are in Europe, you must specify the `eu-west-1` Region by using the `--region eu-west-1` parameter or by setting the `EC2_URL` environment variable.

This section describes how to specify a different region by changing the service endpoint URL.

To specify a different region

1. To view available regions go to [Regions and Endpoints](#) in the *Amazon Web Services General Reference*.
2. If you want to change the service endpoint, set the `EC2_URL` environment variable.

The following example sets `EC2_URL` to the EU (Ireland) Region permanently.

```
C:\> setx EC2_URL https://ec2.eu-west-1.amazonaws.com
```

You're ready to start using Amazon EC2.

Setting Up a Windows HPC Cluster on Amazon Elastic Compute Cloud

This section steps you through how to launch a scalable Microsoft Windows High Performance Computing (HPC) cluster using only Amazon Elastic Compute Cloud (Amazon EC2) instances. A Windows HPC cluster requires an Active Directory domain controller and a DNS server, a head node, and one or more compute nodes. By following the steps in this section, you can assemble each of these components and launch a Windows HPC cluster. For more information on High Performance Computing, go to [High Performance Computing \(HPC\) on AWS](#).

Process for Setting Up a Windows HPC Cluster on Amazon EC2

Task 1: Set Up Your Active Directory Domain Controller (p. 44)
Task 2: Configure Your Head Node (p. 45)
Task 3: Set Up the Compute Node (p. 47)
Task 4: Scale Your HPC Compute Nodes (Optional) (p. 49)

Prerequisites

Before you begin to configure the instances for your Windows HPC cluster, make sure that the following requirements are met:

- Open an Amazon EC2 account, if you haven't already, and set up permissions to launch new EC2 instances. For more information, see [Sign Up for AWS](#).
- Before you begin the configuration in a specific region, check the [Amazon EC2 pricing page](#) and select the drop-down list for that region to see if Cluster Compute Instances are available in that region.
- Install the Amazon EC2 command line tools. For more information, go to [Installing the Amazon EC2 Command Line Tools on Windows \(p. 38\)](#).
- Optionally, you can download the [HPC Pack 2008 R2](#). You can also download HPC Pack 2008 R2 Express directly to your AMI instance later.

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. Setting up your Active Directory involves three steps:

1. Creating security groups for Active Directory.
2. Launching an instance for your domain controller.
3. Configuring your domain controller for your HPC cluster.

Setting Up Security Groups for Active Directory

Run the security group script `create-AD-sec-groups.bat` to create the rules for the domain controller and domain members. If you have not installed the command line tools, manually create a security group with the port requirements for Windows Server 2008/Windows Server 2008 R2. For more information, go to [How to configure a firewall for domains and trusts](#) on the Microsoft website.

To create the required security groups for Active Directory

1. Using a text editor, copy the contents of the [create_AD_security.bat \(p. 50\)](#), and save the file with the name `create-AD-sec-groups.bat` to a computer configured with the Amazon EC2 command line tools from which you connect to Amazon Web Services.
2. Run the file as a local administrator.
3. Log in to the AWS Management Console and verify that the following security groups appear: SG - Domain Controller and SG - Domain Member.

Launch an Instance for Your Domain Controller

Configure your domain controller by launching an instance from AWS and then configuring the instance as a domain controller for your HPC cluster.

To launch an instance for your domain controller

1. Launch an m1.large Amazon EC2 instance type from Microsoft Windows Server 2008 R2 Base (you could use another instance type depending on your anticipated usage) with the name **Domain Controller** and assign it to the **SG - Domain Controller** security group.
2. Create an Elastic IP address and associated this IP address with the Domain Controller instance.
 - a. In the navigation pane, click **Elastic IPs**.
 - b. Click **Allocate New Address**.
 - c. In the **Allocate New Address** dialog box, click **Yes Allocate**.
 - d. Select the Elastic IP address you created, and then click **Associate Address**.
 - e. In the **Associate Address** dialog box, in the **Instance** drop-down list, select the domain controller instance and then click **Yes Associate**.

Configure Your Domain Controller for Your HPC Cluster

Next, 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 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

HPC clients all connect to the head node. The head node facilitates the scheduled jobs. You configure your head node by:

1. Creating security groups for your HPC cluster.
2. Launching an instance for your head node.
3. Installing the HPC Pack.
4. Configuring your cluster.

Creating Security Groups for Your HPC Cluster

Run the security group script `create-HPC-sec-group.bat` to create a security group named **SG - Windows HPC Cluster** with the rules for the HPC cluster nodes. If you have not installed the command line tools, manually create a security group configure with the port requirements for HPC cluster members to communicate only within this security group. For more information, go to [Windows Firewall](#) on the Microsoft website.

To create the required security groups for your HPC cluster

1. Using a text editor, copy the contents of the [create-HPC-sec-group.bat](#) (p. 51), and save the file with the name `create-HPC-sec-group.bat` to a computer configured with the EC2 command line tools from which you connect to Amazon Web Services.
2. Run the file as a local administrator.

3. Log in to AWS Management Console and verify that the security group SG - Windows HPC Cluster appears.

Launch an Instance for the HPC Head Node

Configure your head node by launching a cluster instance from AWS and then configuring the instance as a domain member of the hpc.local and with the necessary user accounts.

To configure an instance for your head node

1. Launch an instance from **Microsoft Windows 2008 R2 64-bit for Cluster Instances** with the name **HPC-Head** and assign the instance to both the **SG - Windows HPC Cluster** and **SG - Domain Member** security groups.
2. Log in to the instance and get the existing DNS server address from **HPC-Head** using **IPConfig /all**.
3. Update the TCP/IPv4 properties of the **HPC-Head** NIC to include the **Domain Controller** Elastic IP address 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 hpc.local\administrator credentials (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-Head** as hpc.local\hpcuser.

Install the HPC Pack

This section explains how to download and 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 2008 R2 Express on **HPC-Head**.
 - a. Download HPC Pack 2008 R2 Express onto **HPC-Head** from <http://go.microsoft.com/fwlink/?LinkID=198084>.
 - b. Extract the files to a folder, open the folder, and double-click **setup.exe**.
 - c. Select **HPC Pack 2008 R2 Express**, and then click **Next**.
 - d. Accept the licensing agreement if you agree, and then click **Next**.
 - e. On the Installation page, select **Create a new HPC cluster by creating a head node**, and then click **Next**.
 - f. Accept the default settings to install all the databases on the Head Node, and then click **Next**.
 - g. Complete the wizard.

Configure Your HPC Cluster on the Head Node

This section explains how to 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**, 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. Launching an instance for your compute node.
2. Installing the HPC Pack on the instance.
3. Adding the compute node to your cluster.

Launch an Instance for the HPC Compute Node

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

To configure an instance for your compute node

1. Launch an instance from **Microsoft Windows 2008 R2 64-bit for Cluster Instances** with the name **HPC-Compute** and assign the instance to both **SG - Windows HPC Cluster** and **SG - Domain Member** security groups.
2. Log in to the instance and get the existing DNS server address from **HPC-Compute** using **IPConfig /all**.
3. Update the TCP/IPv4 properties of the **HPC-Compute** NIC to include the Domain Controller Elastic IP address 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 hpc.local\administrator credentials (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

This section explains how to download and install the HPC Pack on the compute node for your HPC cluster.

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 2008 R2 Express on **HPC-Compute**.
 - a. Download HPC Pack 2008 R2 Express onto **HPC-Compute** from <http://go.microsoft.com/fwlink/?LinkID=198084>.
 - b. Extract the files to a folder, open the folder, and double-click **setup.exe**.
 - c. Select **HPC Pack 2008 R2 Express**, and then click **Next**.
 - d. Accept the licensing agreement if you agree, and then click **Next**.
 - e. On the Installation page, select **Join an existing HPC cluster by creating a new compute node**, and then click **Next**.
 - f. Specify the machine name FQDN of the **HPC-Head** instance, and then choose the defaults.
 - g. 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. Log in to the **HPC-Head** as hpc.local\hpcuser.
2. On **HPC-Head**, open **HPC Cluster Manager**.
3. Select **Node Management** in the bottom-left pane.
4. If the compute node displays in the **Unapproved** bucket, then 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. Log in to **HPC-Compute** as `hpc.local\hpcuser`.
2. Delete any files you downloaded locally from the HP Pack 2008 R2 Express 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. Log in to the AWS Management Console. Click the **Amazon EC2** tab.
8. In **Navigation**, click **Instances**.
9. Wait for the instance status to show **Stopped**.
10. Right-click the instance, and select **Create Image (EBS AMI)**.
11. Specify an image name and image description, and then click **Create This Image** to create an AMI from the instance.
12. Start the original **HPC-Compute** node 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 AWS Management Console, in **Navigation**, click **AMIs**.
16. Use the AMI you created to add additional nodes to the cluster.

**Amazon Elastic Compute Cloud Microsoft Windows
Guide
Running the Lizard Performance Measurement
Application**

Any number of additional compute nodes can now be launched from the AMI that was created. The 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 it directly on your head node as hpc.localhpcuser.

create_AD_security.bat

The following .bat 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"

:: =====
:: 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
```

**Amazon Elastic Compute Cloud Microsoft Windows
Guide
create-HPC-sec-group.bat**

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

:: For Elastic IP to communicate with DNS
call ec2auth %DC% -s 0.0.0.0/0 -P UDP -p 53

:: For RDP for connecting to desktop remotely
call ec2auth %DC% -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 .bat file creates a security group for your HPC cluster nodes. Run this bat file from the client computer from which you are connecting to Amazon Web Services.

```
set HPC="SG - Windows HPC Cluster"

:: =====
:: 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(WS.10).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
```

**Amazon Elastic Compute Cloud Microsoft Windows
Guide
create-HPC-sec-group.bat**

```
call ec2auth %HPC% -o %HPC% -P TCP -p 6729
call ec2auth %HPC% -o %HPC% -P TCP -p 6730
call ec2auth %HPC% -o %HPC% -P TCP -p 7997
call ec2auth %HPC% -o %HPC% -P TCP -p 8677
call ec2auth %HPC% -o %HPC% -P TCP -p 9087
call ec2auth %HPC% -o %HPC% -P TCP -p 9090
call ec2auth %HPC% -o %HPC% -P TCP -p 9091
call ec2auth %HPC% -o %HPC% -P TCP -p 9092
call ec2auth %HPC% -o %HPC% -P TCP -p "9100-9163"
call ec2auth %HPC% -o %HPC% -P TCP -p "9200-9263"
call ec2auth %HPC% -o %HPC% -P TCP -p 9794
call ec2auth %HPC% -o %HPC% -P TCP -p 9892
call ec2auth %HPC% -o %HPC% -P TCP -p 9893
call ec2auth %HPC% -o %HPC% -P UDP -p 9893

:: For HPC related services, these are NOT in the first table but are there in
the third table at link below
:: http://technet.microsoft.com/en-us/library/ff919486\(WS.10\).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% -P TCP -p 3389
```

Glossary

Amazon machine image (AMI)	An Amazon Machine Image (AMI) is an encrypted machine image stored in Amazon S3. It contains all the information necessary to boot instances of your software.
Amazon EBS	A type of storage that enables you to create volumes that can be mounted as devices by Amazon EC2 instances. Amazon EBS volumes behave like raw unformatted external block devices. They have user supplied device names and provide a block device interface. You can load a file system on top of Amazon EBS volumes, or use them just as you would use a block device.
Amazon EBS-backed AMI	An instance launched from an AMI backed by Amazon EBS uses an Amazon EBS volume as its root device. See Amazon EBS .
Instance store-backed AMI	An instance launched from an Amazon S3backed AMI uses an instance store as its root device. See instance store .
Availability Zone	A distinct location within a Region that is engineered to be insulated from failures in other Availability Zones and provides inexpensive, low latency network connectivity to other Availability Zones in the same Region.
compute unit	An Amazon-generated measure that enables you to evaluate the CPU capacity of different Amazon EC2 instance types.
EBS	See Amazon EBS .
Elastic Block Store	See Amazon EBS .
elastic IP address	A static public IP address designed for dynamic cloud computing. Elastic IP addresses are associated with your account, not specific instances. Any elastic IP addresses that you associate with your account remain associated with your account until you explicitly release them. Unlike traditional static IP addresses, however, elastic IP addresses allow you to mask instance or Availability Zone failures by rapidly remapping your public IP addresses to any instance in your account.
ephemeral store	See <i>instance store</i> .
explicit launch permission	Launch permission granted to a specific AWS account.
filter	Criterion you specify to limit the results when you list or describe your EC2 resources.

gibibyte (GiB)	A contraction of giga binary byte, a gibibyte is 2^{30} bytes or 1,073,741,824 bytes. A gigabyte is 10^9 or 1,000,000,000 bytes.
group	See security group .
image	See <i>Amazon machine image</i> .
instance	Once an AMI has been launched, the resulting running system is referred to as an instance. All instances based on the same AMI start out identical and any information on them is lost when the instances are terminated or fail.
instance store	Every instance includes a fixed amount of storage space on which you can store data. This is not designed to be a permanent storage solution. If you need a permanent storage system, use Amazon EBS.
instance type	A specification that defines the memory, CPU, storage capacity, and hourly cost for an instance. Some instance types are designed for standard applications, whereas others are designed for CPU-intensive applications, or memory-intensive applications, etc.
launch permission	AMI attribute allowing AWS accounts to launch an AMI
Linux	Amazon EC2 instances are available for many operating platforms, including Linux, Solaris, Windows, and others.
maximum price	The maximum price you will pay to launch one or more Spot Instances. If your maximum price exceeds the Spot Price and your restrictions are met, Amazon EC2 launches instances on your behalf.
paid AMI	An AMI that you sell to other Amazon EC2 users. For more information, refer to the <i>Amazon DevPay Developer Guide</i> .
private IP address	All Amazon EC2 instances are assigned two IP addresses at launch: a private address (RFC 1918) and a public address that are directly mapped to each other through Network Address Translation (NAT).
public AMI	An AMI that all AWS accounts have launch permissions for.
public data sets	Sets of large 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, like with all AWS services, you pay only for the compute and storage you use for their your applications. These data sets currently include data from the Human Genome Project, the U.S. Census, Wikipedia, and other sources.
public IP address	All Amazon EC2 instances are assigned two IP addresses at launch: a private address (RFC 1918) and a public address that are directly mapped to each other through Network Address Translation (NAT).
region	A geographical area in which you can launch instances (e.g., US, EU).
reservation	A collection of instances started as part of the same launch request.
Reserved Instance	An additional Amazon EC2 pricing option. With Reserved Instances, you can make a low one-time payment for each instance to reserve and receive a significant discount on the hourly usage charge for that instance.
resource	A general term that refers to the objects you work with in Amazon EC2. This includes instances, images, Amazon EBS volumes, snapshots, etc.

security group	A security group is a named collection of access rules. These access rules specify which ingress (i.e., incoming) network traffic should be delivered to your instance. All other ingress traffic will be discarded.
shared AMI	AMIs that developers build and make available for other AWS developers to use.
Solaris	Amazon EC2 instances are available for many operating platforms, including Linux, Solaris, Windows, and others.
snapshot	Amazon EBS provides the ability to create snapshots or backups of your Amazon EBS volumes and store them in Amazon S3. You can use these snapshots as the starting point for new Amazon EBS volumes and to protect your data for long term durability.
Spot Instance	A type of instance that you can bid on to take advantage of unused Amazon EC2 capacity.
Spot Price	The current price for Spot Instances. If your Spot Instance request exceeds this price and your restrictions are met, Amazon EC2 launches instances on your behalf.
supported AMIs	These AMIs are similar to paid AMIs, except that you charge for software or a service that customers use with their own AMIs.
tag	Metadata of your choice (consisting of up to 10 key-value pairs) that you can optionally assign to EC2 resources.
tebibyte (TiB)	A contraction of tera binary byte, a tebibyte is 2^{40} bytes or 1,099,511,627,776 bytes. A terabyte is 10^{12} or 1,000,000,000,000 bytes.
UNIX	Amazon EC2 instances are available for many operating platforms, including Linux, Solaris, Windows, and others.
Windows	Amazon EC2 instances are available for many operating platforms, including Linux, Solaris, Windows, and others.

Document History

This documentation is associated with the 2011-07-15 release of Amazon EC2. This guide was last updated on 15 November 2011.

The following table describes the new and update content in the current release of the *Amazon EC2 Microsoft Windows Guide* documentation set.

Change	Description	Release Date
Updated content	The guide contains a new chapter, Setting Up a Windows HPC Cluster on Amazon Elastic Compute Cloud (p. 43) that explains how to configure a Windows HPC Cluster on Amazon Elastic Compute Cloud and updated instructions for Installing the Amazon EC2 Command Line Tools on Windows (p. 38) .	In this release
Created content	The guide provides information about using Amazon EC2 on Windows instances. For information on the basic infrastructure components of Amazon EC2 on Windows instances, see Introduction to Amazon EC2 (p. 3) . For information on using Windows AMIs, see Using Windows AMIs (p. 11) . For information on setting up your command line interface, see Installing the Amazon EC2 Command Line Tools on Windows (p. 38) .	In this release 23 September 2011

Index

A

AMIs

- paid, 34
- shared, 33
 - finding, 33
 - security, 34
- sharing, 29

B

- batch processing, 3

G

- glossary, 53

I

- introduction, 3

O

- overview, 3

P

- Paid AMIs, 34

R

- Regions, 42

S

- scalable applications, 3
- service overview, 3
- shared AMIs, 33
 - finding, 33
 - security, 34
- sharing AMIs, 29

T

- temporary events, 3

W

- Windows user, 3