
Amazon Elastic Compute Cloud

User Guide

API Version 2009-04-04



Amazon Elastic Compute Cloud: User Guide

Copyright © 2009 Amazon Web Services LLC or its affiliates. All rights reserved.

Table of Contents

Welcome	1
What's New	4
Introduction to Amazon Elastic Compute Cloud	6
Amazon EC2 Concepts	10
AMI and Instance Concepts	10
Amazon EC2 Flow	16
Instance Addressing Concepts	17
Network Security Concepts	18
Region and Availability Zone Concepts	19
Failure Resilient Application Concepts	20
Elastic IP Addresses	20
Auto Scaling	24
Elastic Load Balancing	24
Amazon CloudWatch	24
Public Data Set Concepts	25
Setting Up Amazon EC2	26
Using Amazon EC2	27
Creating and Preparing AMIs	28
Creating an AMI	28
Creating a Linux or UNIX AMI	28
Starting with an Existing AMI	28
Creating an AMI through a Loopback File	34
Creating a Windows AMI	39
Bundling an AMI	44
Bundling a Linux or UNIX AMI	44
Bundling a Windows AMI	47
How to Share AMIs	49
Protecting a Shared AMI (Linux and UNIX)	49
Sharing AMIs	53
How to Make an AMI Public	53
How to Share an AMI with Specific Users	54
How to Publish Shared AMIs	55
Creating Paid AMIs	56
Amazon DevPay and Paid AMIs	56
Product Registration	59
How to Associate a Product Code with an AMI	60
How to Share Your Paid AMI	61
How to Confirm an Instance Is Running with a Product Code	61
How to Get the Product Code from Within an Instance	62
Supported AMIs	62
Launching and Using Instances	64
Instance Metadata	71
Instance Storage	75
Using Shared AMIs	76
Paying for AMIs	78
Getting Console Output and Rebooting Instances	81
Accessing Instances	83
Accessing Instances in Linux and UNIX	83
Accessing Instances in Windows	84
Using Instance Addressing	87
Using Network Security	94
Using Regions and Availability Zones	102
Using Amazon Elastic Block Store	106
Using Auto Scaling, Elastic Load Balancing, and Amazon CloudWatch	115
Auto Scaling	115

Elastic Load Balancing	115
Amazon CloudWatch	115
Enabling Amazon CloudWatch on a New Amazon EC2 Instance	115
Enabling Amazon CloudWatch on an Existing Amazon EC2 Instance	116
Using Public Data Sets	117
Reserving Amazon EC2 Instances	119
Technical FAQ	121
General Information FAQ	121
Operation Information FAQ	122
Instance Types and Architectures FAQ	123
IP Information FAQ	125
Region and Availability Zone FAQ	127
Windows Instances FAQ	129
Monitoring, Errors, and Unexpected Behavior FAQ	129
Reserved Instances FAQs	130
Paid AMIs FAQ	131
Kernels, RAM Disks, and Block Device Mappings FAQ	133
Error Messages FAQ	133
Miscellaneous FAQ	134
Appendix	136
Resources	136
Windows Configuration Service	138
Glossary	140
Document Conventions	143
Index	146

Welcome

Topics

- [Who Should Read This Guide \(p. 1\)](#)
- [Reader Feedback \(p. 2\)](#)
- [How This Guide Is Organized \(p. 2\)](#)
- [Amazon EC2 Resources \(p. 2\)](#)

This is the *Amazon Elastic Compute Cloud User Guide*. This section describes who should read this guide, how the guide is organized, and other resources related to Amazon Elastic Compute Cloud.

The Amazon Elastic Compute Cloud is occasionally referred to within this guide as simply "Amazon EC2"; all copyrights and legal protections still apply.

Who Should Read This Guide

This guide is designed for users that will administer Amazon EC2 instances using the AWS Management Console and the command line tools. This guide picks up where the [Amazon Elastic Compute Cloud Getting Started Guide](#) ends and provides you with the information to create more sophisticated Amazon Machine Images (AMIs), and describes advanced service features.

Required Knowledge and Skills

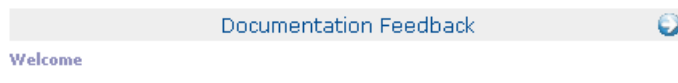
Use of this guide assumes you are familiar with the following:

- XML (For an overview, go to the [W3 Schools XML Tutorial](#))
- Basic understanding of web services (go to [W3 Schools Web Services Tutorial](#))

You should also have worked through the [Amazon Elastic Compute Cloud Getting Started Guide](#), set up the command line tools or AWS Management Console, and have a general understanding of the service.

Reader Feedback

The online version of this guide provides a link at the top of each page that enables you to enter feedback about this guide. We strive to make our guides as complete, error free, and easy to read as possible. You can help by giving us feedback. Thank you in advance!



How This Guide Is Organized

This guide is organized into several major sections described in the following table.

Information	Relevant Sections
Features, common uses, and how we charge for Amazon EC2.	Introduction to Amazon Elastic Compute Cloud (p. 6)
Amazon EC2 concepts and an overview of major Amazon EC2 features.	Amazon EC2 Concepts (p. 10)
Information about how to create a customized software package (operating system and applications) that you can run on Amazon EC2, how to launch instances of the package, and how to access the instances after they launch. Additionally, describes how to use major Amazon EC2 features. These include instance addressing, network security, regions and Availability Zones, Windows, Reserved Instances, and Amazon EBS.	Using Amazon EC2 (p. 27)
Answers to commonly asked questions.	Technical FAQ (p. 121)
Amazon EC2 terms.	Glossary (p. 140)
Typographic and symbol conventions.	Document Conventions (p. 143)

Each section is written to stand on its own, so you should be able to look up the information you need and go back to work. However, you can also read through the major sections sequentially to get in-depth knowledge about Amazon EC2.

Amazon EC2 Resources

The following table lists related resources that you'll find useful as you work with this service.

Amazon Elastic Compute Cloud User Guide
Amazon EC2 Resources

Resource	Description
Amazon Elastic Compute Cloud Getting Started Guide	The Getting Started Guide provides a quick tutorial of the service based on a simple use case. Examples and instructions are included.
Amazon Elastic Compute Cloud User Guide	The Console and Command Line User Guide provides conceptual information about Amazon EC2 and describes how to use Amazon EC2 features using the AWS Management Console and command line tools.
Amazon Elastic Compute Cloud Developer Guide	The Developer Guide provides conceptual information about Amazon EC2 and describes how to use Amazon EC2 features using the SOAP and Query APIs.
Amazon Elastic Compute Cloud API Reference	The API Reference contains a comprehensive description of all SOAP and Query APIs. Additionally, it contains a list of all SOAP data types.
Amazon Elastic Compute Cloud Command Line Reference	The Command Line Tools Reference contains a comprehensive description of all the command line tools and their options.
Amazon EC2 Release Notes	The Release Notes give a high-level overview of the current release. They specifically note any new features, corrections, and known issues.
AWS Developer Resource Center	A central starting point to find documentation, code samples, release notes, and other information to help you build innovative applications with AWS.
Discussion Forums	A community-based forum for developers to discuss technical questions related to Amazon Web Services.
AWS Support Center	The home page for AWS Technical Support, including access to our Developer Forums, Technical FAQs, Service Status page, and AWS Premium Support (if you are subscribed to this program).
AWS Premium Support Information	The primary web page for information about AWS Premium Support, a one-on-one, fast-response support channel to help you build and run applications on AWS Infrastructure Services.
Form for questions related to your AWS account: Contact Us	This form is <i>only</i> for account questions. For technical questions, use the Discussion Forums.
Conditions of Use	Detailed information about the copyright and trademark usage at Amazon.com and other topics.

What's New

This What's New is associated with the 2009-04-04 release of Amazon EC2. This guide was last updated on August 04, 2009.

The following table describes the important changes since the last release of the Amazon EC2 documentation set.

Change	Description	Release Date
Auto Scaling	Auto Scaling enables you to automatically increase or decrease the number of running Amazon EC2 instances in response to your web application's usage and the configuration you define. Auto Scaling makes it easy for you to optimize your Amazon EC2 usage, automatically scaling your cluster to ensure your application has the right number of instances running to meet your workload demands. Auto Scaling is particularly well suited for applications that experience hourly, daily, or weekly variability in usage. For more information, see Amazon Auto Scaling Developer Guide .	18 May 2009
Elastic Load Balancing	Elastic Load Balancing offers the ability to evenly spread requests across your running Amazon EC2 instances. Unlike traditional load balancers or load balancing software, there is no need to provision, manage, or plan for load balancing capacity needs. Each Elastic Load Balancer is automatically scaled, fully fault-tolerant, and distributes incoming application traffic across a group of Amazon EC2 instances. For more information, see Elastic Load Balancing Developer Guide .	18 May 2009

Change	Description	Release Date
Amazon CloudWatch	<p>Amazon CloudWatch is a monitoring service for Amazon EC2 that is designed to gather, aggregate, store, and retrieve metrics. Amazon CloudWatch makes it easy to monitor your Amazon EC2 instances and aggregate metrics from instances like CPU or disk utilization over different time ranges and across different pools of resources. This service is tightly integrated with Amazon EC2's Auto Scaling and Elastic Load Balancing, enabling you to use monitoring metrics to trigger scaling activities.</p> <p>For more information, see Amazon CloudWatch Developer Guide.</p>	18 May 2009
New Guides	<p>Amazon EC2 now consists of six guides:</p> <ul style="list-style-type: none"> • Amazon Elastic Compute Cloud Getting Started Guide—Describes how to set up your environment and get started with Amazon EC2. • Amazon Elastic Compute Cloud User Guide—Describes Amazon EC2 concepts and how to use Amazon EC2 with the AWS Management Console or the command line tools. • Amazon Elastic Compute Cloud Developer Guide—Describes Amazon EC2 concepts and how to use Amazon EC2 with the APIs. • Amazon Elastic Compute Cloud API Reference—Provides detailed information about the Amazon EC2 APIs. • Amazon Elastic Compute Cloud Command Line Reference—Provides detailed information about the Amazon EC2 command line tools.. • Amazon Elastic Compute Cloud Quick Reference Card—Provides a quick summary of the Amazon EC2 command line tools. 	18 May 2009

Introduction to Amazon Elastic Compute Cloud

Topics

- [What Is Amazon EC2? \(p. 6\)](#)
- [Advantages of Amazon EC2 \(p. 6\)](#)
- [Popular Uses for Amazon EC2 \(p. 8\)](#)
- [Amazon EC2 Charges \(p. 8\)](#)

What Is Amazon EC2?

Amazon EC2 is a web service that enables you to launch and manage server instances in Amazon's data centers using APIs or available tools and utilities. You can use Amazon EC2 server instances at any time, for as long as you need, and for any legal purpose. If you need 100 instances for a two-day research project, sure. If you need a group of instances that can be scaled up and down to meet the traffic fluctuations of your Facebook application, no problem.

Instances are available in different sizes and configurations. This allows us to provide different instance types that you can use to meet specific needs. For example, you might want to use an m1.small instance (one Amazon EC2 Compute Unit) as a web server, an m1.xlarge instance (eight Amazon EC2 Compute Units) as a database server, or an extra large High-CPU instance (twenty Amazon EC2 Compute Units) for processor intensive applications.

What makes Amazon EC2 different is that you use only the capacity that you need. This eliminates your need to make large and expensive hardware purchases, reduces the need to forecast traffic, and enables you to immediately deal with changes in requirements or spikes in popularity related to your application or service.

Advantages of Amazon EC2

- **Elastic**—Amazon EC2 enables you to increase or decrease capacity within minutes, not hours or days. You can commission one, hundreds or even thousands of server instances simultaneously. Of course, because this is all controlled with web service APIs, your application can automatically scale itself up and down depending on its needs.

- **Completely Controlled**—You have complete control of your instances. You have root access to each one, and you can interact with them as you would any machine. Instances can be rebooted remotely using web service APIs. You also have access to console output of your instances.
- **Flexible**—You have the choice of several instance types, allowing you to select a configuration of memory, CPU, operating system, and instance storage that is optimal for your application.
- **Designed for use with other Amazon Web Services**—Amazon EC2 works in conjunction with Amazon Simple Storage Service (Amazon S3), Amazon SimpleDB and Amazon Simple Queue Service (Amazon SQS) to provide a complete solution for computing, query processing and storage across a wide range of applications.
- **Reliable**—Amazon EC2 offers a highly reliable environment where replacement instances can be rapidly and reliably commissioned. The service runs within Amazon’s proven network infrastructure and data centers.
- **Multiple Locations**—Amazon EC2 provides the ability to place instances in multiple locations. Amazon EC2 locations are composed of Regions and Availability Zones. Availability Zones are distinct locations that are engineered to be insulated from failures in other Availability Zones and provide inexpensive, low latency network connectivity to other Availability Zones in the same Region. By launching instances in separate Availability Zones, you can protect your applications from failure of a single location. Regions consist of one or more Availability Zones, are geographically dispersed, and will be in separate geographic areas or countries.
- **Secure**—Amazon EC2 provides web service interfaces to configure firewall settings that control network access to and between groups of instances.
- **Inexpensive**—Amazon EC2 passes on to you the financial benefits of Amazon’s scale. You pay a very low rate for the compute capacity you actually consume.
 - **On-Demand Instances**—On-Demand Instances let you pay for compute capacity by the hour with no long-term commitments. This frees you from the costs and complexities of planning, purchasing, and maintaining hardware and transforms what are commonly large fixed costs into much smaller variable costs. On-Demand Instances also remove the need to buy “safety net” capacity to handle periodic traffic spikes.
 - **Reserved Instances**—Reserved Instances give you the option to make a low, one-time payment for each instance you want to reserve and in turn receive a significant discount on the hourly usage charge for that instance. After the one-time payment for an instance, that instance is reserved for you, and you have no further obligation; you may choose to run that instance for the discounted usage rate for the duration of your term, or when you do not use the instance, you will not pay usage charges on it.

Features for Building Failure Resilient Applications

- **Amazon Elastic Block Store**—Amazon Elastic Block Store (Amazon EBS) offers persistent storage for Amazon EC2 instances. Amazon EBS volumes provide off-instance storage that persists independently from the life of an instance. Amazon EBS volumes are highly available, highly reliable volumes that can be attached to a running Amazon EC2 instance and are exposed as standard block devices. Amazon EBS volumes offer greatly improved durability over local Amazon EC2 instance stores, as Amazon EBS volumes are automatically replicated on the backend (in a single Availability Zone). For those wanting even more durability, Amazon EBS provides the ability to create point-in-time consistent snapshots of your volumes that are then stored in Amazon S3, and automatically replicated across multiple Availability Zones. These snapshots can be used as the starting point for new Amazon EBS volumes, and can protect your data for long term durability.
- **Elastic IP Addresses**—Elastic IP addresses are static IP addresses designed for dynamic cloud computing. An Elastic IP address is associated with your account not a particular instance, and you control that address until you choose to explicitly release it. Unlike traditional static IP addresses, however, Elastic IP addresses allow you to mask instance or Availability Zone failures by programmatically remapping your public IP addresses to any instance in your account. Rather than waiting on a data technician to reconfigure or replace your host, or waiting for DNS to propagate to all of your customers, Amazon EC2 enables you to engineer around problems with your instance or software by quickly remapping your Elastic IP address to a replacement instance.

- **Auto Scaling**—Auto Scaling offers the ability to automatically increase or decrease the number of running Amazon EC2 instances in response to your web application's usage and the configuration you define. Auto Scaling makes it easy for you to optimize your Amazon EC2 usage, automatically scaling your cluster to ensure your application has the right number of instances running to meet your workload demands. Auto Scaling is particularly well suited for applications that experience hourly, daily, or weekly variability in usage.
- **Elastic Load Balancing**—Elastic Load Balancing offers the ability to evenly spread requests across your running Amazon EC2 instances. Unlike traditional load balancers or load balancing software, there is no need to provision, manage, or plan for load balancing capacity needs. Each Elastic Load Balancer is automatically scaled, fully fault-tolerant, and distributes incoming application traffic across a group of Amazon EC2 instances.
- **Amazon CloudWatch**—Amazon CloudWatch is a monitoring service for Amazon EC2 that is designed to gather, aggregate, store, and retrieve metrics. Amazon CloudWatch makes it easy to monitor your Amazon EC2 instances and aggregate metrics from instances like CPU or disk utilization over different time ranges and across different pools of resources. This service is tightly integrated with Amazon EC2's Auto Scaling and Elastic Load Balancing, enabling you to use monitoring metrics to trigger scaling activities.

Popular Uses for Amazon EC2

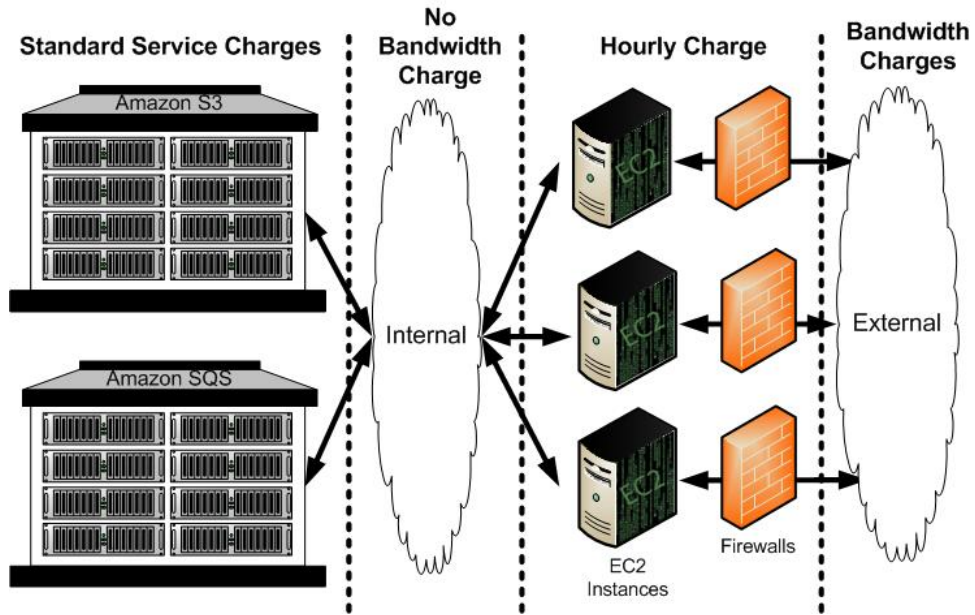
Although the applications for Amazon EC2 are only limited by your ingenuity, the following is a list of popular uses for Amazon EC2:

- **Scalable Applications**—You can build a scalable application that shrinks or expands to meet your current demands.
This can help you use only the compute resources that you need and can help you respond to events where a mention on a popular news site can result in a dramatic spike in traffic.
- **Temporary Events**—You can use Amazon EC2 for temporary solutions and one-off events that would require you to maintain compute resources that are normally idle.
This includes hosting conferences in virtual worlds, live blogging, distribution of newly released media, and short-term promotional web sites.
- **Batch Processing**—You can use Amazon EC2 for projects that require massive compute resources which would be expensive to build on your own.
This includes video and image processing, financial data processing, and science and research applications.
- **Fault Resilient Applications**—You can build an application across multiple availability zones which will be protected against the loss of an entire physical location.

Amazon EC2 Charges

With Amazon EC2, you don't have to pay upfront fees, you don't have to commit to a fixed amount of bandwidth, and you don't have to meet any minimum usage requirements. As with other AWS services, you only pay for what you use.

The following figure summarizes how you are charged for using Amazon EC2.



For detailed information on Amazon EC2 charges, go to the [Amazon EC2 Product Page](#).

Amazon EC2 Concepts

Topics

- [AMI and Instance Concepts \(p. 10\)](#)
- [Amazon EC2 Flow \(p. 16\)](#)
- [Instance Addressing Concepts \(p. 17\)](#)
- [Network Security Concepts \(p. 18\)](#)
- [Region and Availability Zone Concepts \(p. 19\)](#)
- [Failure Resilient Application Concepts \(p. 20\)](#)
- [Public Data Set Concepts \(p. 25\)](#)

This section describes concepts you should understand before using Amazon EC2

AMI and Instance Concepts

This section describes AMIs and instances, the basic building blocks of Amazon EC2. Before accomplishing anything with Amazon EC2, you must understand the concepts in this section.

AMIs

An Amazon Machine Image (AMI) is an encrypted machine image that contains all information necessary to boot instances of your software. For example, an AMI might contain Linux, Apache, and your web site or it might contain Linux, Hadoop, and a custom application.

AMIs are stored in Amazon S3.

Public AMIs are made available by Amazon and the Amazon EC2 community and can be downloaded from the Resource Center. You can use public AMIs as a base to create your own custom private AMIs.

Private AMIs are AMIs that you own and can only be accessed by you or those to whom you grant access.

Paid AMIs are AMIs that you purchase from developers or AMIs that come with service contracts from organization such as Red Hat.

Shared AMIs are AMIs that developers build and make available for other AWS developers to use. Building safe, secure, useable AMIs for public consumption is a fairly straightforward process, if you follow a few simple guidelines. For information on how to use shared AMIs and how to share AMIs, see [Using Shared AMIs \(p. 76\)](#) and [How to Share AMIs \(p. 49\)](#).

Bundling an AMI

To use a file system image with Amazon EC2, you must bundle it as an AMI. The bundling process does the following:

- 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
- Creates a manifest file that contains a list of the image parts with their checksums

Instances

After an AMI is launched, the resulting running system is called an instance. By default, you can run up to 20 instances. If you need more than 20 instances, please complete the [Amazon EC2 Instance Request Form](#) and your request will be considered.

Instances remain running unless they fail or are terminated. When this happens, the data on the instance is no longer available.

Instance Usage

The instance is your basic computation building block. Amazon EC2 offers multiple instance types from which you can choose. You can run as many or as few instances as you need at any given time.

For information about available instance types, see [Instance Types \(p. 12\)](#).

Once launched, an instance looks very much like a traditional host. You have complete control of your instances; you have root access to each one and you can interact with them as you would any machine.

Here are some suggestions for making the best use of Amazon EC2 instances:

- Do not rely on an instance's local storage for valuable, long-term data.
When instances fail, the data on the local disk is lost. Use a replication strategy across multiple instances to keep your data safe or store your persistent data in Amazon S3
- Define images based on the type of work they perform.
For "Internet applications," you might define one image for database instances and another for web servers. Image creation and storage are cheap and easy operations, so you can individualize and customize as necessary. Specialized images can result in smaller AMI sizes, which boot considerably faster.
- Monitor the health of your instances.
You can make your instances work for you by configuring them to monitor each other. For example, you could create an image that contains a monitoring tool (e.g., the open-source Nagios or OpenNMS on Linux and UNIX). Then, your other instances could report their health to the monitoring instance.
- Keep your Amazon EC2 firewall permissions as restrictive as possible.
Only open up permissions that you require. Use separate *groups* to deal with instances that have different security requirements. Consider using additional security measures inside your instance (such as using your own firewall). If you need to log in interactively (ssh), consider creating a bastion

security group that allows external login and keep the remainder of your instances in a group that does not allow external login.

Instance Types

Amazon EC2 instances are grouped into two families: standard and High-CPU. Standard instances have memory to CPU ratios suitable for most general purpose applications; High-CPU instances have proportionally more CPU resources than memory (RAM) and are well suited for compute-intensive applications. When selecting instance types, you might want to use less powerful instance types for your web server instances and more powerful instance types for your database instances. Additionally, you might want to run CPU instance types for CPU-intensive data processing tasks.

One of the advantages of EC2 is that you pay by the instance hour, which makes it convenient and inexpensive to test the performance of your application on different instance families and types. One good way to determine the most appropriate instance family and instance type is to launch test instances and benchmark your application.

Available Instance Types

The instance types described in the following table are available.

Type	CPU	Memory	Storage	Platform	I/O	Name
Small	1 EC2 Compute Unit (1 virtual core with 1 EC2 Compute Unit)	1.7 GB	160 GB instance storage (150 GB plus 10 GB root partition)	32-bit	Moderate	m1.small
Large	4 EC2 Compute Units (2 virtual cores with 2 EC2 Compute Units each)	7.5 GB	850 GB instance storage (2 x 420 GB plus 10 GB root partition)	64-bit	High	m1.large
Extra Large	8 EC2 Compute Units (4 virtual cores with 2 EC2 Compute Units each)	15 GB	1690 GB instance storage (4 x 420 GB plus 10 GB root partition)	64-bit	High	m1.xlarge
High-CPU Medium	5 EC2 Compute Units (2 virtual cores with 2.5 EC2 Compute Units each)	1.7 GB	350 GB instance storage (340 GB plus 10 GB root partition)	32-bit	Moderate	c1.medium
High-CPU Extra Large	20 EC2 Compute Units (8 virtual cores with 2.5 EC2 Compute Units each)	7 GB	1,690 GB instance storage (4 x 420 GB plus 10 GB root partition)	64-bit	High	c1.xlarge



Note

The *small* instance type is the original Amazon EC2 instance type available since the launch of Amazon EC2. It is the default instance type for all customers. To use other instance types, you must specify them through the `RunInstances` operation.



Important

We strongly recommend using the 2.6.18 Xen stock kernel with the c1.medium and c1.xlarge instances. Although the default Amazon EC2 kernels work, the new kernels provide greater stability and performance for these instance types. For more information about kernels, see [Kernels, RAM Disks, and Block Device Mappings FAQ \(p. 133\)](#).

Instance Storage

Every instance includes a fixed amount of storage space on which you can store data. Within this document, it is referred to as the "instance store" as it is not designed to be a permanent storage solution.

If an instance reboots (intentionally or unintentionally), the data on the instance store will survive. If the underlying drive fails or the instance is terminated, the data will be lost.

We highly recommend backing up important data to Amazon S3.

Storage Locations

Storage is exposed on the instance types as described in the following table.

Location	Description
/dev/sda1	Formatted and mounted as root (/) on all Linux and UNIX instance types. Formatted and mounted as C:\ on all Windows instance types.
/dev/sda2 or xvdb (Windows)	Formatted and mounted as /mnt on m1.small and c1.medium instances. Formatted and mounted on small Windows instance types.
/dev/sda3	Formatted and mounted as /swap on m1.small and c1.medium instances on all Linux and UNIX instance types. Not available on Windows instances.
/dev/sdb or xvdb (Windows)	Formatted and mounted as /mnt on m1.large, m1.xlarge, and c1.xlarge Linux and UNIX instances. Formatted and mounted on m1.large, m1.xlarge, and c1.xlarge Windows instances.
/dev/sdc or xvdc (Windows)	Available on m1.large, m1.xlarge, and c1.xlarge Linux and UNIX instances. Formatted and mounted on m1.large, m1.xlarge, and c1.xlarge Windows instances.
/dev/sdd or xvdd (Windows)	Available on m1.xlarge and c1.xlarge Linux and UNIX instances. Formatted and mounted on m1.xlarge and c1.xlarge Windows instances.
/dev/sde or xvde (Windows)	Available on m1.xlarge and c1.xlarge Linux and UNIX instances. Formatted and mounted on m1.xlarge and c1.xlarge Windows instances.

On-Demand and Reserved Instances

This section describes the differences between standard On-Demand and Reserved Instances.

On-Demand Instance Concepts

On-Demand Instances let you pay for compute capacity by the hour with no long-term commitments. This frees you from the costs and complexities of planning, purchasing, and maintaining hardware and transforms what are commonly large fixed costs into much smaller variable costs.



Note

For information about pricing, refer to the [Amazon EC2 Product Page](#).

Reserved Instance Concepts

With Amazon EC2 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.

Amazon EC2 Reserved Instances are based on instance type and location (region and Availability Zone) for a specified period of time (e.g., 1 year or 3 years) and are only available for Linux/UNIX instances.



Note

For information about pricing, refer to the [Amazon EC2 Product Page](#).

For information on using Reserved Instances, see [Reserving Amazon EC2 Instances \(p. 119\)](#).

How Reserved Instances are Applied

Reserved Instances are applied to instances that meet the type/location criteria during the specified period. In this example, a user is running the following instances:

- (4) m1.small instances in Availability Zone us-east-1a
- (4) c1.medium instances in Availability Zone us-east-1b
- (2) c1.xlarge instances in Availability Zone us-east-1b

The user then purchases the following Reserved Instances.

- (2) m1.small instances in Availability Zone us-east-1a
- (2) c1.medium instances in Availability Zone us-east-1a
- (2) m1.xlarge instances in Availability Zone us-east-1a

Amazon EC2 applies the two m1.small Reserved Instances to two of the instances in Availability Zone us-east-1a. Amazon EC2 doesn't apply the two c1.medium Reserved Instances because the c1.medium instances are in a different Availability Zone and does not apply the m1.xlarge Reserved Instances because there are no running m1.xlarge instances.

Windows Instance Types

This section describes major concepts that you should understand when using Windows instances.

Differences Between Windows and Linux/UNIX Instances

Using Amazon EC2 instances running Windows is similar to using instances running Linux and UNIX. The following are the major differences between instances that use Linux/UNIX and Windows:

- **Remote Desktop**—To access Windows instances, you use Remote Desktop instead of SSH.
- **Administrative Password**—To access Windows instances the first time, you must obtain the administrative password using the `ec2-get-password` command.
- **Simplified Bundling**—To bundle a Windows instance, you use a single command that shuts down the instance, saves it as an AMI, and restarts it.

Amazon EC2 Running Windows

As part of this service, Amazon EC2 instances can now run Microsoft Windows Server 2003. Our base Windows image provides you with most of the common functionality associated with Windows. However, if you require more than two concurrent Windows users or need to leverage applications

that require LDAP, Kerberos, RADIUS, or other credential services, you must use Windows with Authentication Services. For example, Microsoft Exchange Server and Microsoft SharePoint Server require Windows with Authentication Services.



Note

To get started using Windows instances, we recommend using the AWS Management Console.

There are differences in pricing between Windows and Windows with Authentication Services instances. For information on pricing, go to the [Amazon EC2 Product Page](#).

Windows AMI

Amazon EC2 currently provides the following Windows AMIs:

- Windows Authenticated (32-bit)
- Windows Authenticated (64-bit)
- Windows Anonymous (32-bit)
- Windows Anonymous (64-bit)

The Windows public AMIs that Amazon provides are unmodified versions of Windows with the following two exceptions: we added drivers to improve the networking and disk I/O performance and we created the Amazon EC2 configuration service. The Amazon EC2 configuration service performs the following functions:

- Randomly sets the Administrator password on initial launch, encrypts the password with the user's SSH key, and reports it to the console. This operation happens upon initial AMI launch. If you change the password, AMIs that are created from this instance use the new password.
- Configures the computer name to the internal DNS name. To determine the internal DNS name, see [Using Instance Addressing \(p. 87\)](#).
- Sends the last three system and application errors from the event log to the console. This helps developers to identify problems that caused an instance to crash or network connectivity to be lost.

Measuring Compute Resources

Transitioning to a utility computing model changes how developers are trained to think about CPU resources. Instead of purchasing or leasing a particular processor to use for several months or years, you are renting capacity by the hour. Because Amazon EC2 is built on commodity hardware, over time there might be several different types of physical processors underlying different virtual EC2 instances. Our goal is to provide a consistent amount of CPU capacity regardless of the actual underlying hardware.

Amazon EC2 uses a variety of measures to provide each instance with a consistent and predictable amount of CPU capacity. To make it easy for developers to compare CPU capacity between different instance types, we defined an Amazon EC2 Compute Unit.



Note

We use several internal benchmarks and tests to manage the consistency and predictability of the performance of an Amazon EC2 Compute Unit. For more information, go to the [Instance page](#).

To find out which instance works best for your application, we recommend launching an instance and using your own benchmark application. This helps you determine which instance type works best for your specific use case.

I/O Resources

Amazon EC2 provides virtualized server instances. While some resources like CPU, memory and instance storage are dedicated to a particular instance, other resources like the network and the disk subsystem are shared amongst instances. If each instance on a physical host tries to use as much of one of these shared resources as possible, each receives an equal share of that resource. However, when a resource is under-utilized you are often able to consume a higher share of that resource while it is available.

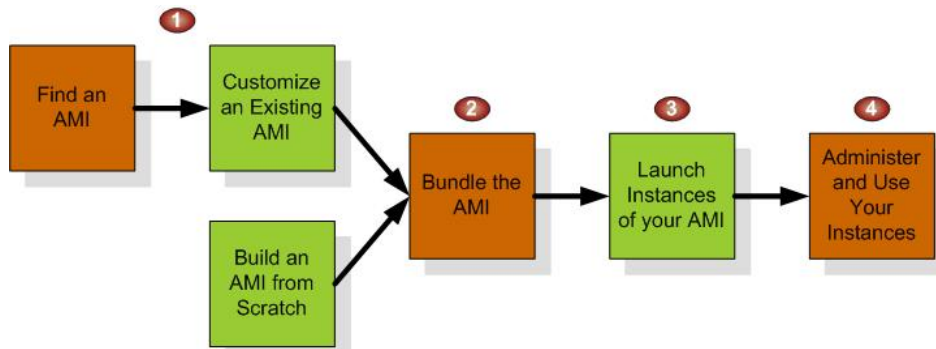
The different instance types provide higher or lower minimum performance from the shared resources depending on their size. Each of the instance types has an I/O performance indicator (moderate or high). Instance types with high I/O performance have a larger allocation of shared resources. Allocating larger share of shared resources also reduces the variance of I/O performance. For most applications, moderate I/O performance is more than enough. However, for applications that require greater or more consistent I/O performance, consider instances with high I/O performance.

Related Topics

- [Creating and Preparing AMIs \(p. 28\)](#)
- [Launching and Using Instances \(p. 64\)](#)
- [Accessing Instances \(p. 83\)](#)

Amazon EC2 Flow

The following graphic and table explain the basic flow for using Amazon EC2.



Launch Confirmation Process

1	You create an AMI from scratch (Linux and UNIX only) or based on an existing AMI. This is optional, as you can launch instances of existing AMIs without modifying them. See Creating an AMI (p. 28) .
2	You bundle the AMI and obtain an AMI ID so you can launch as many instances of the AMI as you want. See Bundling an AMI (p. 44) .
3	You launch one or more instances of your AMI. See Launching and Using Instances (p. 64) .
4	You administer and use your instances as you would with any servers.

Related Topics

- [Creating and Preparing AMIs \(p. 28\)](#)
- [Launching and Using Instances \(p. 64\)](#)
- [Accessing Instances \(p. 83\)](#)

Instance Addressing Concepts

This section describes the types of IP addresses available to Amazon EC2 instances, including elastic IP addresses that can be remapped on demand.

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). Private addresses are only reachable from within the Amazon EC2 network. Public addresses are reachable from the Internet.

Amazon EC2 also provides an internal DNS name and a public DNS name which map to the private and public IP addresses respectively. The internal DNS name can only be resolved within Amazon EC2. The public DNS name resolves to the public IP address outside the Amazon EC2 network and the private IP address within the Amazon EC2 network.



Note

If you require persistent Internet routable IP addresses that can be assigned to and removed from instances as necessary, use elastic IP addresses. For more information, see [Elastic IP Addresses \(p. 20\)](#).

Private (RFC 1918) Addresses

All Amazon EC2 instances are allocated a private address by DHCP. These ranges are defined in RFC 1918, are only routable within Amazon EC2, and are used for communication between instances. For more information, go to [RFC 1918](#).

This private address is associated exclusively with the instance for its lifetime and is only returned to Amazon EC2 when the instance terminates.

Always use the internal address when you are communicating between Amazon EC2 instances. This ensures that your network traffic follows the highest bandwidth, lowest cost, and lowest latency path through our network.

Internal DNS Name

Each instance is provided an internal DNS name that resolves to the private IP address of the instance from within Amazon EC2; it will not resolve outside of Amazon EC2.

Public Addresses

At launch, a public address is also associated with each Amazon EC2 instance using Network Address Translation (NAT). For more information about NAT, go to [RFC 1631: The IP Network Address Translator \(NAT\)](#).

This public address is associated exclusively with the instance until it is terminated or replaced with an elastic IP address.



Important

Amazon EC2 instances that access other instances through their public NAT IP address are charged for regional or Internet data transfer, depending on whether the instances are in the same region.

Public DNS

Each instance is provided an external DNS name that resolves to the public IP address of the instance outside the Amazon EC2 network and the private IP address from within Amazon EC2 network.

Related Topics

- [Elastic IP Addresses \(p. 20\)](#)
- [Using Instance Addressing \(p. 87\)](#)

Network Security Concepts

The Amazon EC2 service allows you to dynamically add and remove instances. However, this flexibility can complicate firewall configuration and maintenance which traditionally relies on IP addresses, subnet ranges or DNS host names as the basis for the firewall rules.

The Amazon EC2 firewall allows you to assign your instances to user-defined *groups* and define firewall rules for these groups. As instances are added or removed, the appropriate rules are enforced. Similarly, if you change a rule for a group, the changes are automatically applied to all members of the group.

Security Groups

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.

You can modify rules for a group at any time. The new rules are automatically enforced for all running instances and instances launched in the future.



Note

You can create up to 100 security groups.

Group Membership

When you launch an AMI instance, you can assign it to as many groups as you like.

If no groups are specified, the instance is assigned to the `default` group. By default, this group allows all network traffic from other members of this group and discards traffic from other IP addresses and groups. If this does not meet your needs, you can modify the rule settings of the `default` group.



Note

After an instance is running, the security groups to which it belongs cannot be changed.

Group Access Rights

The access rules define source based access either for named security groups or for IP addresses (i.e., CIDR-based rules). For CIDR-based rules, you can also specify the protocol and port range (or ICMP type and code).

Related Topics

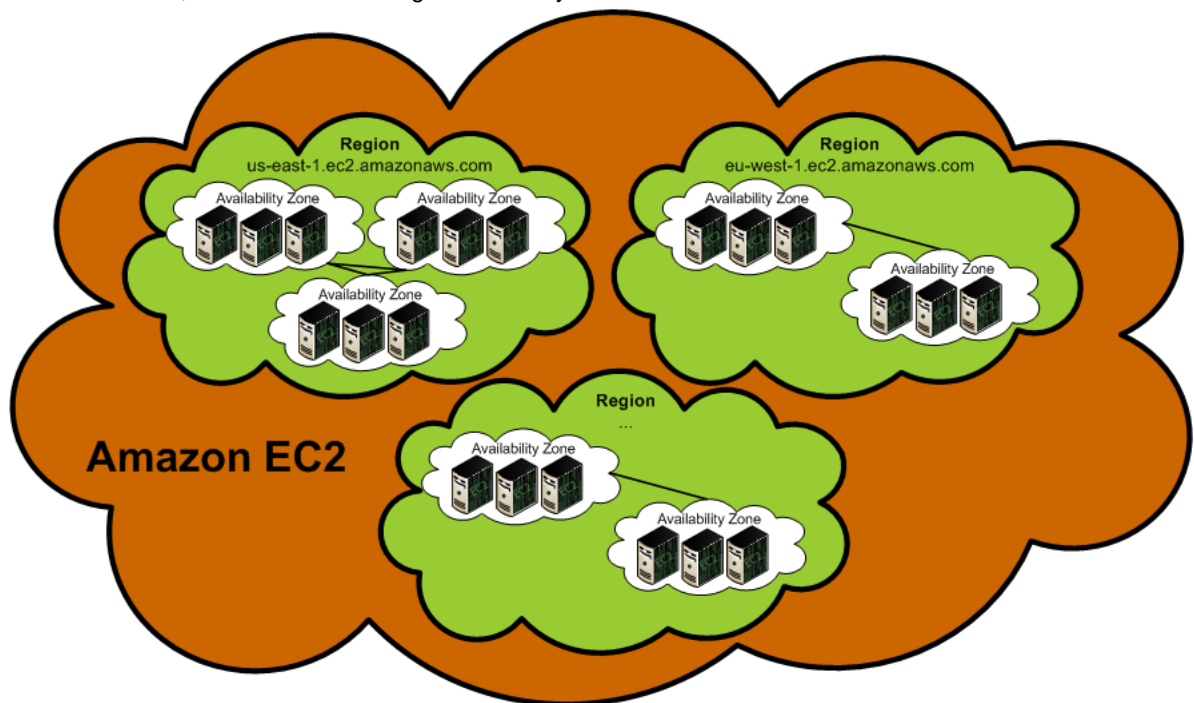
- [Using Network Security \(p. 94\)](#)

Region and Availability Zone Concepts

Amazon EC2 provides the ability to place instances in multiple locations. Amazon EC2 locations are composed of Availability Zones and regions. Regions are dispersed and located in separate geographic areas (e.g., US and EU). Availability Zones are distinct locations within a region that are engineered to be isolated from failures in other Availability Zones and provide inexpensive, low latency network connectivity to other Availability Zones in the same region.

By launching instance in separate regions, you can design your application to be closer to specific customers or to meet legal or other requirements. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location.

The following graphic shows Amazon EC2. Each region is completely independent. Each Availability Zone is isolated, but connected through low-latency links.



Regions

Amazon EC2 provides multiple regions so you can launch Amazon EC2 instances in locations that meet your requirements. For example, you might want to launch instances in Europe to be closer to your European customers or to meet legal requirements.

Each Amazon EC2 region is designed to be completely isolated from the other Amazon EC2 regions. This achieves the greatest possible failure independence and stability, and it makes the locality of each EC2 resource unambiguous.

To launch or work with instances, you must specify the correct regional URL endpoint. For example, to access the United States region, you make service calls to the `us-east-1.ec2.amazonaws.com` service endpoint. To access the region in Europe, you make service calls to the `eu-west-1.ec2.amazonaws.com` service endpoint.

Availability Zones

Amazon operates state-of-the-art, highly available data center facilities. However, failures can occur that affect the availability of instances that are in the same location. Although this is rare, if you host all your Amazon EC2 instances in a single location that is affected by such a failure, your instances will be unavailable.

For example, if you have instances distributed across three Availability Zones and one of them fails, you can design your application so the instances in the remaining Availability Zones handle any requests.



Note

You can use Availability Zones in conjunction with elastic IP addresses to remap IP addresses across Availability Zones. For information on elastic IP addresses, see [Elastic IP Addresses](#) (p. 20).

Related Topics

- [Using Regions and Availability Zones](#) (p. 102)
- [Region and Availability Zone FAQ](#) (p. 127)

Failure Resilient Application Concepts

Elastic IP Addresses

By default, all Amazon EC2 instances are assigned two IP addresses at launch: a private (RFC 1918) address and a public address that is mapped to the private IP address through Network Address Translation (NAT).

If you use dynamic DNS to map an existing DNS name to a new instance's public IP address, it might take up to 24 hours for the IP address to propagate through the Internet. As a result, new instances might not receive traffic while terminated instances continue to receive requests.

To solve this problem, Amazon EC2 provides elastic IP addresses. Elastic IP addresses are static IP addresses 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.



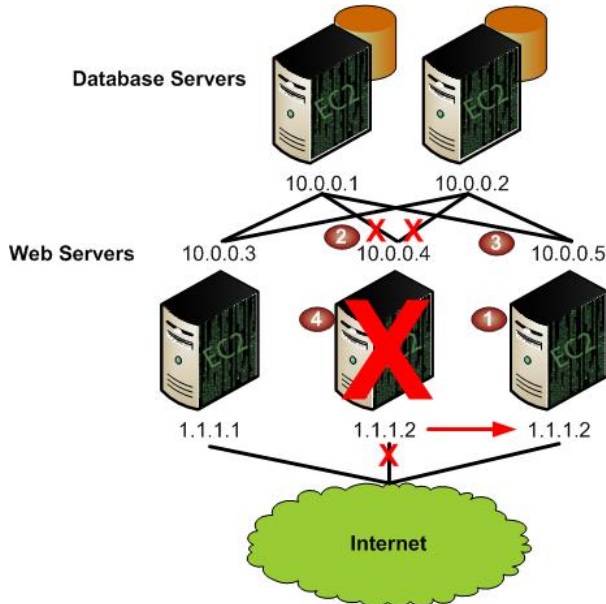
Note

You can only associate one elastic IP address with one instance at a time.

When you associate an elastic IP address with an instance, its current public IP address is released to the Amazon EC2 public IP address pool. If you disassociate an elastic IP address

from the instance, the instance is automatically assigned a new public IP address within a few minutes.

In the following image, web servers are connected to the Internet through elastic IP addresses and to database servers through their private IP addresses.



The administrator decides to replace a web server with a larger instance type. To do this, the administrator starts a new instance using a larger instance type (1), disassociates an elastic IP address from a running instance (2), associates the elastic IP address with the new instance (3), and terminates the old instance (4).



Note

To ensure our customers are efficiently using elastic IP addresses, we impose a small hourly charge when these IP addresses are not mapped to an instance. When these IP addresses are mapped to an instance, they are free of charge.

Amazon Elastic Block Store

Amazon Elastic Block Store (Amazon EBS) is a type of storage designed specifically for Amazon EC2 instances. Amazon EBS allows 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.

You can create up to twenty Amazon EBS volumes of any size (from one GiB up to one TiB). Each Amazon EBS volume can be attached to any Amazon EC2 instance in the same Availability Zone or can be left unattached. If you need more than 20 volumes, please complete the [Amazon EBS Volume Limit Request Form](#) and your request will be considered.

Amazon EBS provides the ability to create snapshots (backups) of your Amazon EBS volumes to Amazon S3. You can use these snapshots as the starting point for new Amazon EBS volumes and can protect your data for long term durability.

Amazon EBS volumes provide the following:

- Off-instance storage
- Persistence beyond the lifetime of instances
- High availability and reliability
- Ability to attach to and detach from a running instance
- Exposure as a device within an instance

Amazon EBS snapshots provide the following:

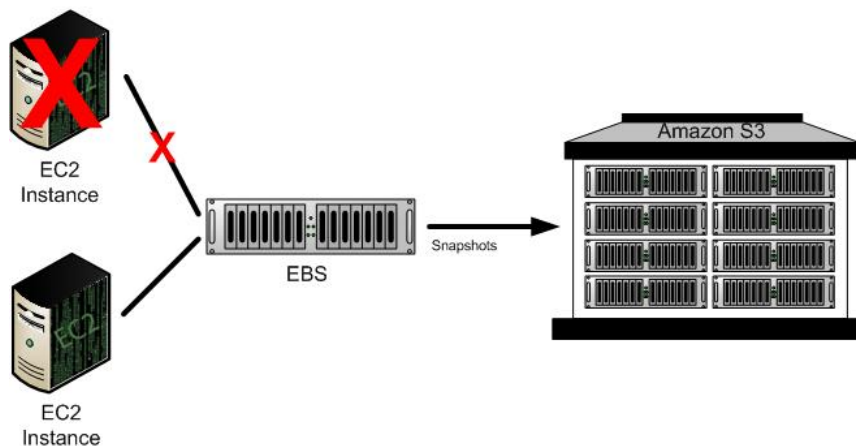
- Ability to capture the current state of a volume
- Data backup
- A method for instantiating new volumes that contain the exact contents of a snapshot

Amazon EBS Use Cases

This section describes common Amazon EBS use cases.

Fault Tolerance

Amazon EBS is designed to allow you to attach any instance to a storage volume. In the event you experience an instance failure, your Amazon EBS volume automatically detaches with your data intact. You can then reattach the volume to a new instance and quickly recover.

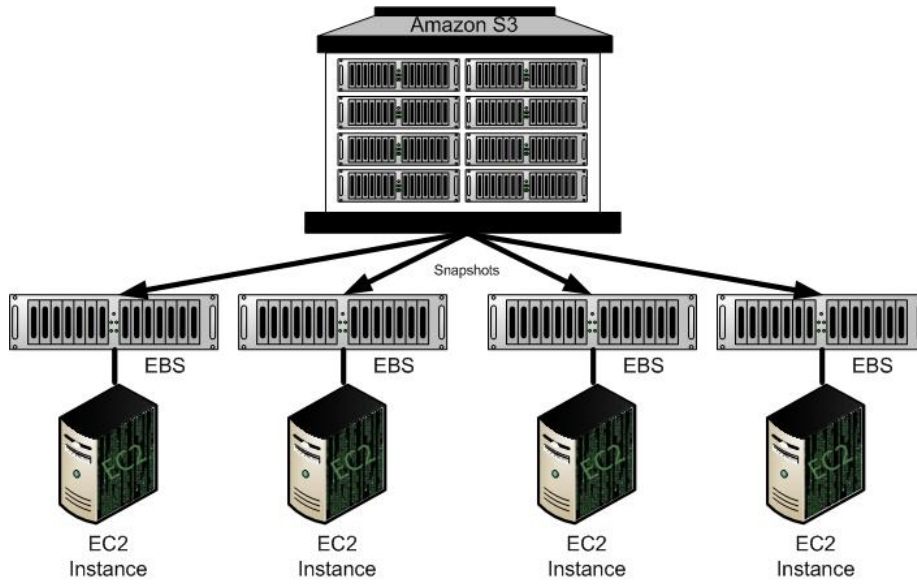


Launch Process

1	You are running an Amazon EC2 instance that is attached to an Amazon EBS volume, when your Amazon EC2 instance fails or is experiencing problems.
2	To recover, you detach the Amazon EBS volume from your instance (if it has not already automatically detached), launch a new Amazon EC2 instance, and attach the Amazon EBS volume to the new instance.
3	In the unlikely event the Amazon EBS volume fails, you can create a new Amazon EBS volume based on the most recent snapshot of your failed volume.

Launching New Volumes Using Snapshots

Amazon EBS snapshots allow you to quickly launch new volumes, using a previous snapshot as a starting point for those new volumes.



Launch Process

1	You are running a web service with a large data set.
2	When the data is ready, you can store a snapshot of your volume to Amazon S3 for long term durability.
3	When traffic and resource usage increases, you can launch a new volume from the snapshot, launch a new instance, and then attach the new volume to the new instance.
4	If traffic decreases, you can shut down one or more Amazon EC2 instances and delete their Amazon EBS volumes.

Data Persistence

Amazon EBS volumes exist separately from the actual instances and persist until you delete them. This allows you to store your data without leaving an Amazon EC2 instance running.

Launch Process

1	You run an instance periodically to perform a batch processing job on a large and growing data set.
2	At the end of your job, you shut down the Amazon EC2 instance, but leave your Amazon EBS volume running.
3	The next time you process the data set, you launch a new Amazon EC2 instance and reattach it to your existing Amazon EBS volume.

Using this model, you can process and store your data set indefinitely, only using the processing and storage resources that you require

Large Data Sets

Amazon EBS offers larger volumes than provided by Amazon EC2 instances. Each Amazon EBS volume can be up to one TiB Amazon EBS in size.

Related Topics

- [Using Amazon Elastic Block Store \(p. 106\)](#)

Auto Scaling

Auto Scaling enables you to scale up or down the number of instances you are using based on parameters that you specify, such as traffic or CPU load.

Auto Scaling also monitors the health of each Amazon EC2 instance that it launches. If any instance terminates unexpectedly, Auto Scaling detects the termination and launches a replacement instance.

For a high degree of flexibility, you can organize Amazon EC2 instances into `AutoScalingGroups`, which enable you to scale different server classes (e.g., web servers, back end servers) at different rates. For each group, you specify the minimum number of instances, the maximum number of instances, and the parameters to increase and decrease the number of running instances.

For more information, refer to the *Amazon Auto Scaling Developer Guide*.

Elastic Load Balancing

Elastic Load Balancing lets you automatically distribute the incoming traffic (or load) among all the instances you are running. The service also makes it easy to add new instances when you need to increase the capacity of your web site application.

Customers reach your web site via your web URL, such as `www.mywebsite.com`. This single address might actually represent several instances of your running web application. To always have an available web site, you need to run multiple instances. Otherwise, your customers might see delays when accessing your site, or worse, might not be able to access your site at all.

Elastic Load Balancing manages the incoming requests by optimally routing traffic so that no one instance is overwhelmed. You can quickly add more instances to applications that are experiencing an upsurge in traffic or remove capacity when traffic is slow.

For more information, refer to the *Elastic Load Balancing Developer Guide*.

Amazon CloudWatch

Amazon CloudWatch collects raw data from partnered AWS services such as Amazon EC2 and then processes the information into readable, near real-time metrics. These statistics are recorded for a period of two weeks, allowing you access to historical information and providing you with a better perspective on how your web application or service is performing.

Amazon CloudWatch runs a monitoring services that collects raw measurement data or *measures*, such as `CPUUtilization` (percentage of Amazon EC2 compute units used by an instance) or `DiskWriteBytes` (number of bytes written in a minute). Measures consist of a name (e.g., `DiskWriteBytes`), a value (e.g., 0 bytes per second), additional metadata that provides more context, and a timestamp.

For more information, refer to the *Amazon CloudWatch Developer Guide*.

Related Topics

- [Using Amazon Elastic Block Store \(p. 106\)](#)
- [Auto Scaling \(p. 115\)](#)

- [Elastic Load Balancing](#) (p. 115)
- [Amazon CloudWatch](#) (p. 115)

Public Data Set Concepts

Amazon EC2 provides a repository of public data sets that can be seamlessly integrated into AWS cloud-based applications. Amazon stores the data sets at no charge to the community and, like all AWS services, users pay only for the compute and storage they use for their own applications.

Previously, large data sets such as the mapping of the Human Genome and the US Census data required hours or days to locate, download, customize, and analyze. Now, anyone can access these data sets from their Amazon EC2 instances and start computing on the data within minutes. Users can also leverage the entire AWS ecosystem and easily collaborate with other AWS users. For example, users can produce or use prebuilt server images with tools and applications to analyze the data sets. By hosting this important and useful data with cost-efficient services such as Amazon EC2, AWS hopes to provide researchers across a variety of disciplines and industries with tools to enable more innovation, more quickly.



Note

For more information, go to the [Public Data Sets Page](#)

Available Public Data Sets

Public data sets are currently available in the following categories:

API List

- **Biology**—Includes Human Genome Project, GenBank, and other content.
- **Chemistry**—Includes multiple versions of PubChem and other content.
- **Economics**—Includes census data, labor statistics, transportation statistics, and other content.
- **Encyclopedic**—Includes Wikipedia content from multiple sources and other content.

Related Topics

- [Using Public Data Sets](#) (p. 117)

Setting Up Amazon EC2

Before you can follow the procedures in [Using Amazon EC2 \(p. 27\)](#), you must set up your Amazon EC2 environment for use with the AWS Management Console or the command line tools. To do so, follow the procedures in the [Amazon Elastic Compute Cloud Getting Started Guide](#).

Using Amazon EC2

Topics

- [Creating and Preparing AMIs \(p. 28\)](#)
- [Launching and Using Instances \(p. 64\)](#)
- [Accessing Instances \(p. 83\)](#)
- [Using Instance Addressing \(p. 87\)](#)
- [Using Network Security \(p. 94\)](#)
- [Using Regions and Availability Zones \(p. 102\)](#)
- [Using Amazon Elastic Block Store \(p. 106\)](#)
- [Using Auto Scaling, Elastic Load Balancing, and Amazon CloudWatch \(p. 115\)](#)
- [Using Public Data Sets \(p. 117\)](#)
- [Reserving Amazon EC2 Instances \(p. 119\)](#)

This section contains procedures that describe how to create, launch, and access AMIs, as well as how to use major Amazon EC2 features.



Note

For detailed information the command line tools, go to the [Amazon Elastic Compute Cloud Command Line Reference](#). For detailed information about the APIs, go to the [Amazon Elastic Compute Cloud API Reference](#)

Creating and Preparing AMIs

Topics

- [Creating an AMI \(p. 28\)](#)
- [Bundling an AMI \(p. 44\)](#)
- [How to Share AMIs \(p. 49\)](#)
- [Creating Paid AMIs \(p. 56\)](#)

This section describes how to build, store, and share Amazon Machine Images (AMIs).

Creating an AMI

Topics

- [Creating a Linux or UNIX AMI \(p. 28\)](#)
- [Creating a Windows AMI \(p. 39\)](#)

Creating a Linux or UNIX AMI

Topics

- [Starting with an Existing AMI \(p. 28\)](#)
- [Creating an AMI through a Loopback File \(p. 34\)](#)

In Linux and UNIX, there are two common ways to create an AMI that offer a mix of ease of use and detailed customization levels.

The easiest method involves starting from an existing [public AMI](#) and modifying it according to your requirements, as described in [Starting with an Existing AMI \(p. 28\)](#).

Another approach is to build a fresh installation either on a stand-alone machine or on an empty file system mounted by loopback. This essentially entails building an operating system installation from scratch and is described in [Creating an AMI through a Loopback File \(p. 34\)](#).

After the installation package has been built to your satisfaction, you must bundle it and upload it to Amazon Simple Storage Service (Amazon S3) as described in [Bundling an AMI \(p. 44\)](#).



Note

Creating a Linux or UNIX AMI requires you to download and install the AMI tools in addition to the API tools. For more information, refer to the [Amazon Elastic Compute Cloud Getting Started Guide](#).

This section provides detailed instructions on creating an AMI. For information on quickly launching an existing AMI, go to the [Amazon Elastic Compute Cloud Getting Started Guide](#).

Starting with an Existing AMI

To quickly and easily get a new working AMI, start with an existing public AMI or one of your own. You can then modify it and create a new AMI with the `ec2-bundle-vol` utility described in [Bundling an AMI \(p. 44\)](#).



Note

Before selecting an AMI, determine whether the instance types you plan to launch are 32-bit or 64-bit. For more information, see [Instance Types \(p. 12\)](#)

Make sure you are using GNU Tar 1.15 or later.

To use an existing AMI to create a new AMI, complete the following tasks.

Tasks to Use an Existing AMI

1	How to Select an AMI (p. 29)
2	How to Generate a Key Pair (p. 29)
3	How to Launch the Instance (p. 31)
4	How to Authorize Network Access (p. 32)
5	How to Connect to the Instance (p. 32)
6	How to Upload the Key and Certificate (p. 33)

How to Select an AMI

First, locate an AMI that contains the packages and services you require. This can be one of your own AMIs or a public AMI provided by Amazon EC2.

To select an AMI

1. Get a list of available AMIs by entering the `ec2-describe-images` command:

```
$ ec2-describe-images -a
```

The response includes the image ID, the location of the file in Amazon S3, and whether the file is available.

2. Choose an AMI from the list and write down its AMI ID.

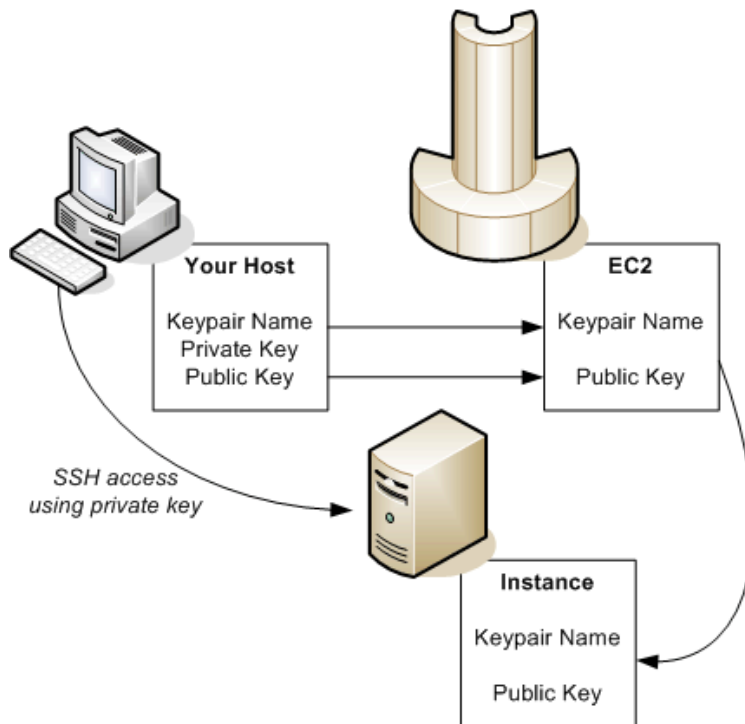
Example

```
$ ec2-describe-images -o self -o amazon
IMAGE ami-60a54009 ec2-public-images/base-fc4-apache.manifest.xml
 475219833042 available public
IMAGE ami-61a54028 <your-s3-bucket>/image.manifest.xml AIDADH4IGTRXXKCD
available private
IMAGE ami-2bb65342 ec2-public-images/getting-started.manifest.xml
 475219833042 available public
IMAGE ami-6ea54007 ec2-public-images/base-fc3-mysql.manifest.xml 475219833042
available public
```

How to Generate a Key Pair

This task is only required if you selected one of the public AMIs provided by Amazon EC2. You must create a public/private key pair to ensure that only you have access to instances that you launch.

After you generate a key pair, the public key is stored in Amazon EC2 using the key pair name you selected. Whenever you launch an instance using the key pair name, the public key is copied to the instance metadata. This allows you to access the instance securely using your private key.



To create a public/private key pair

1. Enter the following command:

```
$ ec2-add-keypair <keypair-name>
```

The <keypair-name> is the name you select for the key pair.

The resulting private key is displayed.

2. Open a text editor.
3. Paste the entire private key, starting with the line "-----BEGIN RSA PRIVATE KEY-----" and ending with the line "-----END RSA PRIVATE KEY-----".
4. Save the file and exit.



Note

This file should only be readable by the file owner.

Example

```
$ ec2-add-keypair gsg-keypair
KEYPAIR gsg-keypair
 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f
-----BEGIN RSA PRIVATE KEY-----
MIIEoQIBAAKCAQBULFg5ujHrtmljnutSuoO8Xe56LlT+HM8v/xkaa39EstM3/aFxTHgElQiJLChp
HungXQ29VTc8rc1bw0lkdi23OH5eqkMHGhvEwqa0HWASUM1l4o3o/IX+0f2UcPoKCOVUR+jx7lSg
5AU52EQfanIn3ZQ8lFW7Edp5a3q4DhJgLUkToHVbicL5E+g45zfB95wIyywWZfEW/UUF3LpGZyq/
ebIUlq1qTbHkLbCC2r7RTn8vpQWp47BGVYgtGSBMPTRP5hnbzZuqj3itkiLHjU39S2sJcJ0TrJx5
i8BygR4s3mHKBj8l+ePQxG1kGbF6R4Yg6sECmXn17MRQVXODNHZbAgMBAAECggEAY1tsiUsIwDl5
91CXirkYGUvFLyLflXenxfI50mDFms/mumTqloHO7tr0oriHDR5K7wMcY/YY5YkcXNo7mvUVDlpM
ZNUJs7rw9gZRTrf7LylaJ58kOcyajw8TsC4e4LPbFaHwSld6K8rXh64o6WgW4SrsB6ICmrlkGQI7
3wcfgt5ecIu4TZf00E9IHjn+2eRlSrjBdeORi7KiUNC/pAG23I6MdDOFEQRcCSigCj+4/mciFUSA
SWS4dMbrpb9FNSIcf9dcLxVM7/6KxgJNfZc9XWzUw77Jg8x92Zd0fVhHOux5IZC+UvSKWB4dyfcI
tE8C3p9bbU9VGyY5vLCAiIb4qQKBgQDLiO24GXrIkswF32YtBBMuVgLGcWU9h9HlO9mKAc2m8Cml
jUE5IpzrjTcdc9I2qiIMUTwtgnw42auSCzbUeYMRPtdqyQ7p6AjMujp9EPemcSVOK9vXYL0PtcO
xW9MC0dtV6iPkCN7gOqiZXPRKaFbWADp16p8UAIvS/a5XXk5jwKBgQCKkPHi2EiShluRkxhljyWC
iDCiK6JBRsMvpLbc0v5dKwP5a1o1fmdR5PJaV2qvZSj5CYNpMay1/EDNTY5OSIJU+0KfMQbyhsbm
rdLNLDL4+TcnT7c62/aH01ohYaf/VcBRhtLlBfGqQc7+sAc8vmKkesnF7CqCEKdyF/dhrxYdQKB
gC0iZzzNAapayzl+JcVTwweid6j9JqNXbBc+Z2YwMi+T0Fv/P/hwkX/ypeOXnIUcw0Ih/YtGBVAC
DQbsz7LcYlHqXiHKYNWNvXgwwO+oiChjxvEkSdsTTIfnk4VScvU9BxDbQHjdiNDJbL6oar92UN7V
rBYvChJZF7LvUH4YmVpHAoGAbZ2X7XvoeEO+uZ58/BGKOlGHByHBDiXtZMhdJr15HTYjxK7OgTZm
gK+8zp4L9IbvLGDmJ08vft32XPEWuvI8twCzFH+CsWLQADZMZKSSBasOZ/h1FwhdMgCMcY+Qlzd4
JZKjTSu3i7vhvx6RzdSedXEMNTZWN4qlIx3kR5aHcucGyYA9T+ZrvmlF0seQPbLknn7EqhXIjBaT
P8TTvW/6bdPi23ExzxZn7KodrfclYRph1LHMPaONv/x2xALIf91UB+v5ohy1oDoasL0gi1jhouRe
2ERKKdWz0ZL9SWq6VTdhr/5G994CK72fy5WhyERbdjUIDHaK3M849JJuf8cSrvSb4g==
-----END RSA PRIVATE KEY-----
```

How to Launch the Instance

You are now ready to launch an instance of the AMI that you previously selected.

To launch an instance

1. Start the launch by entering the following command:

```
$ ec2-run-instances <ami_id> -k <keypair-name>
```

The `<ami_id>` is the AMI ID you selected earlier and `<keypair-name>` is the name of the key pair. The command will return the AMI instance ID, a unique identifier for each launched instance. You use the instance ID to manipulate the instance. This includes viewing the status of the instance, terminating the instance, and so on.

Launching the instance will take a few minutes.

2. View the progress of the instance by entering the following command:

```
$ ec2-describe-instances <instance_id>
```

The `<instance_id>` is the ID of the instance.

When the status field displays "running," the instance was created and is booting. However, the instance might not be immediately accessible over the network. Make sure to use the appropriate DNS name provided by the `ec2-describe-instances` command.



Important

Once you launch an instance, you will be billed for all usage, including hourly CPU time. Make sure to terminate any instances that you do not want to leave running. For information on Amazon EC2 pricing, go to the [Amazon EC2 home page](#).

Example

The following example launches an instance of ami-2bb65342.

```
$ ec2-run-instances ami-2bb65342 -k gsg-keypair
RESERVATION    r-302dc059      416161254515    default
INSTANCE      i-eb977f82      ami-2bb65342    pending gsg-
keypair       0    m1.small    2007-10-16T07:56:20+0000    us-east-1a
```

The following shows the status of the launch:

```
$ ec2-describe-instances i-eb977f82
RESERVATION    r-302dc059      416161254515    default
INSTANCE      i-eb977f82      ami-2bb65342
ec2-72-44-40-222.compute-1.amazonaws.com    10-251-50-83.ec2.internal
running gsg-keypair    0    m1.small    2007-10-16T07:56:20+0000    us-
east-1a
```

How to Authorize Network Access

To authorize access to your instance

1. Enter the `ec2-authorize` command to allow all IP addresses to access your instance through the port 80 (public web) IP address.

```
$ ec2-authorize default -p 80
PERMISSION    default    ALLOWS    tcp    80    80    FROM    CIDR
0.0.0.0/0
```

2. Get the public IP address of your local machine by going to a search engine, entering "what is my IP address," and using one of the provided services.
3. Enter the `ec2-authorize` command to open port 22 (SSH port) to your IP address.

```
$ ec2-authorize default -p 22 -s your_ip_address/32
PERMISSION    default    ALLOWS    tcp    22    22    FROM    CIDR
your_ip_address/32
```

This command allows access from your IP address only. If your IP address is dynamic, you will need to use this command each time it changes. To allow additional IP address ranges, use this command for each range.

How to Connect to the Instance

After starting an instance, you can log in and modify it according to your requirements.

To connect to an instance

- If you are launching an AMI that supports SSH login (e.g., public AMIs), use the following command to log in with your private key:

```
$ ssh -i <private-keyfile> root@<dns_location>
```

The `<private-keyfile>` is the file that contains the private key and `dns_location` is the DNS location of the instance within Amazon EC2. Your instance displays a prompt that contains your username and the hostname of the instance.

You now have complete control over the instance. You can add, remove, modify, or upgrade packages and files to suit your needs.



Important

We recommend exercising extreme care when changing some of the basic Amazon EC2 configuration settings, such as the network interface configuration and the `/etc/fstab` contents. Otherwise, the AMI might become unbootable or inaccessible from the network once running.

Example

The following example shows logging in to an AMI using SSH.

```
$ ssh -i id_rsa-gsg-keypair  
root@ec2-67-202-51-223.compute-1.amazonaws.com  
root@ec2-67-202-51-223 #
```

How to Upload the Key and Certificate

Your new AMI is encrypted and signed to ensure that only you and Amazon EC2 can access it. Therefore, you must upload your Amazon EC2 private key and X.509 certificate to the running instance, for use in the AMI bundling process.



Note

For information on obtaining your Amazon EC2 private key and X.509 certificate, refer to the [Amazon Elastic Compute Cloud Getting Started Guide](#).

To upload your Amazon EC2 private key and X.509 certificate

1. Copy your Amazon EC2 private key and X.509 certificate to the `/mnt` directory.
2. Enter the following command:

```
$ scp <private_keyfile> <certificate_file> root@<dns_location>:/mnt
```

The `<private_keyfile>` is the file that contains the private key, `certificate_file` is the file that contains the certificate, and `dns_location` is the DNS location of the instance within Amazon EC2.

Amazon EC2 returns the name of the files and some performance statistics.



Note

It is important that the key and cert files are uploaded into `/mnt` to prevent them from being bundled with the new AMI.

You are ready to bundle the volume and uploading the resulting AMI to Amazon S3. For more information, see [Bundling an AMI \(p. 44\)](#).

Example

```
$ scp pk-HKZYKTAIG2ECMXIYBH3HXV4ZBZQ55CLO.pem  
cert-HKZYKTAIG2ECMXIYBH3HXV4ZBZQ55CLO.pem  
root@ec2-67-202-51-223.compute-1.amazonaws.com:/mnt  
-i id_rsa-gsg-keypair  
pk-HKZYKTAIG2ECMXIYBH3HXV4ZBZQ55CLO.pem 100% 717 0.7KB/s 00:00  
cert-HKZYKTAIG2ECMXIYBH3HXV4ZBZQ55CLO.pem  
100% 685 0.7KB/s 00:00
```

Creating an AMI through a Loopback File

This method involves doing a full operating system installation on a clean root file system, but avoids having to create a new root disk partition and file system on a physical disk. Once you have installed your operating system, the resulting image can be bundled as an AMI with the `ec2-bundle-image` utility.



Note

Before selecting an AMI, determine whether the instance types you plan to launch are 32-bit or 64-bit. For more information, see [Instance Types \(p. 12\)](#)

Make sure you are using GNU Tar 1.15 or later.

These examples use Fedora Core 4. Please make any adjustments for your distribution.

AMI Creation Process

1	How to Create a File to Host the AMI (p. 34).
2	How to Create a Root File System inside the File (p. 34).
3	How to Mount the File through Loopback (p. 35).
4	How to Prepare for the Installation (p. 36).
5	How to Install the Operating System (p. 37).
6	How to Configure the Operating System (p. 38).

How to Create a File to Host the AMI

The `dd` utility can create files of arbitrary sizes. Make sure to create a file large enough to host the operating system, tools, and applications that you will install. For example, a baseline Linux and UNIX installation requires about 700MB, so your file should be at least 1 GB.

To create a file to host the AMI

- Enter the following command:

```
# dd if=/dev/zero of=image_name bs=1M count=size
```

The `<image_name>` is the name of the image file you are creating and `<size>` is the size of the file in megabytes.

Example

The following command creates a one gigabyte file (1024*1MB).

```
# dd if=/dev/zero of=my-image.fs bs=1M count=1024
1024+0 records in
1024+0 records out
```

How to Create a Root File System inside the File

There are several variations on the `mkfs` utility that can create a file system inside the image file you are creating. Typical Linux and UNIX installations default to `ext2` or `ext3` file systems.

To create an `ext3` file system

- Enter the following command:

```
# mke2fs -F -j <image_name>
```

The *<image_name>* is the name of the image file.

Example

The following command creates an `ext3` file system.

```
# mke2fs -F -j my-image.fs
mke2fs 1.38 (30-Jun-2005)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
131072 inodes, 262144 blocks
13107 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=268435456
8 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376
```

```
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

This filesystem will be automatically checked every 24 mounts or 180 days, whichever comes first. Use `tune2fs -c` or `-i` to override.

How to Mount the File through Loopback

The loopback module allows you to use a normal file as if it were a raw device, which gives you a file system within a file. Mounting a file system image file through loopback presents it as part of the normal file system. You can then modify it using your favorite file management tools and utilities.

To mount the file through loopback

1. Create a mount point in the file system where the image will be attached:

```
# mkdir <image_mountpoint>
```

The *<image_mountpoint>* is the location where the image will be mounted.

2. Mount the file system image:

```
# mount -o loop <image_name> <image_mountpoint>
```

The *<image_name>* is the name of the image file and *<image_mountpoint>* is the mount location.

Example

The following commands create and mount the `my-image.fs` image file.

```
# mkdir /mnt/ec2-fs
# mount -o loop my-image.fs /mnt/ec2-fs
```

How to Prepare for the Installation

Before the operating system installation can proceed, you must create and prepare the newly created root file system.

To prepare for the installation

1. Create a `/dev` directory and populate it with a minimal set of devices. You can ignore the errors in the output.

```
# mkdir /mnt/ec2-fs/dev
# /sbin/MAKEDEV -d <image_mountpoint>/dev -x console
# /sbin/MAKEDEV -d <image_mountpoint>/dev -x null
# /sbin/MAKEDEV -d <image_mountpoint>/dev -x zero
```

The `<image_mountpoint>` is the mount location.

2. Create the `fstab` file within the `/etc` directory and add the following:

```
/dev/sda1 /          ext3      defaults        1 1
none      /dev/pts  devpts    gid=5,mode=620 0 0
none      /dev/shm  tmpfs     defaults        0 0
none      /proc     proc      defaults        0 0
none      /sys      sysfs     defaults        0 0
```

3. Create a temporary yum configuration file (e.g., `yum-xen.conf`) and add the following content.

```
[main]
cachedir=/var/cache/yum
debuglevel=2
logfile=/var/log/yum.log
exclude=-debuginfo
gpgcheck=0
obsoletes=1
reposdir=/dev/null

[base]
name=Fedora Core 4 - $basearch - Base
mirrorlist=http://fedora.redhat.com/download/mirrors/fedora-core-
$releasever
enabled=1

[updates-released]
name=Fedora Core 4 - $basearch - Released Updates
mirrorlist=http://fedora.redhat.com/download/mirrors/updates-released-fc
$releasever
enabled=1
```

This ensures all the required basic packages and utilities are installed. This file can be located anywhere on your main file system (not on your loopback file system) and is only used during installation.

4. Enter the following:

```
# mkdir <image_mountpoint>/proc
# mount -t proc none <image_mountpoint>/proc
```

The `<image_mountpoint>` is the mount location. A `groupadd` utility bug in the `shadow-utils` package (versions prior to 4.0.7-7) requires you to mount the new `proc` file system manually with the preceding command.

Example

These commands create the `/dev` directory and populate it with a minimal set of devices:

```
# mkdir /mnt/ec2-fs/dev
# /sbin/MAKEDEV -d /mnt/ec2-fs/dev -x console
MAKEDEV: mkdir: File exists
MAKEDEV: mkdir: File exists
MAKEDEV: mkdir: File exists
# /sbin/MAKEDEV -d /mnt/ec2-fs/dev -x null
MAKEDEV: mkdir: File exists
MAKEDEV: mkdir: File exists
MAKEDEV: mkdir: File exists
# /sbin/MAKEDEV -d /mnt/ec2-fs/dev -x zero
MAKEDEV: mkdir: File exists
MAKEDEV: mkdir: File exists
MAKEDEV: mkdir: File exists
```

This example creates and mounts the `/mnt/ec2-fs/proc` directory.

```
# mkdir /mnt/ec2-fs/proc
# mount -t proc none /mnt/ec2-fs/proc
```

How to Install the Operating System

At this stage, the basic directories and files are created and you are ready to install the operating system. Depending on the speed of the host and network link to the repository, this process might take a while.

To install the operating system

- Enter the following command:

```
# yum -c <yum_configuration_file> --installroot=<image_mountpoint> -y
  groupinstall Base
```

The `<yum_configuration_file>` is the name of the yum configuration file and `<image_mountpoint>` is the mount location.

You now have a base installation, which you can configure for operation inside Amazon EC2 and customize for your use.

Example

This example installs the operating system at the `/mnt/ec2-fs` mount point using the `yum-xen.conf` yum configuration file.

```
# yum -c yum-xen.conf --installroot=/mnt/ec2-fs -y groupinstall Base
Setting up Group Process
Setting up repositories
base                100% |=====| 1.1 kB    00:00
updates-released   100% |=====| 1.1 kB    00:00
comps.xml           100% |=====| 693 kB    00:00
comps.xml           100% |=====| 693 kB    00:00
Setting up repositories
Reading repository metadata in from local files
primary.xml.gz      100% |=====| 824 kB    00:00
base                : ##### 2772/2772
Added 2772 new packages, deleted 0 old in 15.32 seconds
primary.xml.gz      100% |=====| 824 kB    00:00
updates-re: ##### 2772/2772
Added 2772 new packages, deleted 0 old in 10.74 seconds
...
Complete!
```

How to Configure the Operating System

After successfully installing the base operating system, you must configure the networking and hard drives to work in the Amazon EC2 environment.

To configure the operating system

1. Edit (or create) `/mnt/ec2-fs/etc/sysconfig/network-scripts/ifcfg-eth0` and make sure it contains at least the following information:

```
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
TYPE=Ethernet
USERCTL=yes
PEERDNS=yes
IPV6INIT=no
```



Note

The Amazon EC2 DHCP server ignores hostname requests. If you set `DHCP_HOSTNAME`, the local hostname will be set on the instance but not externally. Additionally, the local hostname will be the same for all instances of the AMI, which might be confusing.

2. Ensure that networking starts by making sure the following line appears in the `/mnt/ec2-fs/etc/sysconfig/network` file:

```
NETWORKING=yes
```

3. Ensure that local disk storage on `/dev/sda2` and swap space on `/dev/sda3` are mounted at system startup by adding the following lines to `/mnt/ec2-fs/etc/fstab`:

```
/dev/sda2 /mnt      ext3    defaults    0 0
/dev/sda3 swap        swap    defaults    0 0
```



Note

The `/dev/sda2` and `/dev/sda3` storage locations only apply to small instances. For more information on instance storage, see [Instance Storage \(p. 13\)](#).

4. Make sure all of your required services start at system startup by allocating them appropriate system run levels. For example, to enable the service `my-service` on multi-user and networked run levels, enter the following commands:

```
# chroot /mnt/ec2-fs /bin/sh
# chkconfig --level 345 my-service on
# exit
```

Your new installation is successfully installed and configured to operate in the Amazon EC2 environment.

5. Umount the image by entering the following commands:

```
# umount <image_mountpoint>/proc
# umount -d <image_mountpoint>
```

The `<image_mountpoint>` is the mount location.

Example

The following example unmounts the installation from the `/mnt/ec2-fs` mount point.

```
# umount /mnt/ec2-fs/proc
# umount -d /mnt/ec2-fs
```

Creating a Windows AMI

This section describes and provides instructions on how to create an AMI in Windows.



Note

Before selecting an AMI, determine whether the instance types you plan to launch are 32-bit or 64-bit. For more information, see [Instance Types \(p. 12\)](#)

To create an AMI using Windows, complete the following tasks.

Tasks to Use an Existing AMI

1	How to Select an AMI (p. 39)
2	How to Generate a Key Pair (p. 40)
3	How to Launch the Instance (p. 41)
4	How to Get the Administrator Password (p. 42)
5	How to Authorize Network Access (p. 43)
6	How to Connect to the Instance (p. 43)
7	How to Load Software and Make Changes (p. 44)

How to Select an AMI

First, locate an AMI that contains the packages and services you require. This can be one of your own AMIs, a public AMI provided by Amazon EC2, or a public AMI provided by a Amazon EC2 developer or user.

To select an AMI

1. To get a list of available AMIs, enter the `ec2-describe-images` command:

```
C:\> ec2-describe-images -o self -o amazon | findstr /i windows

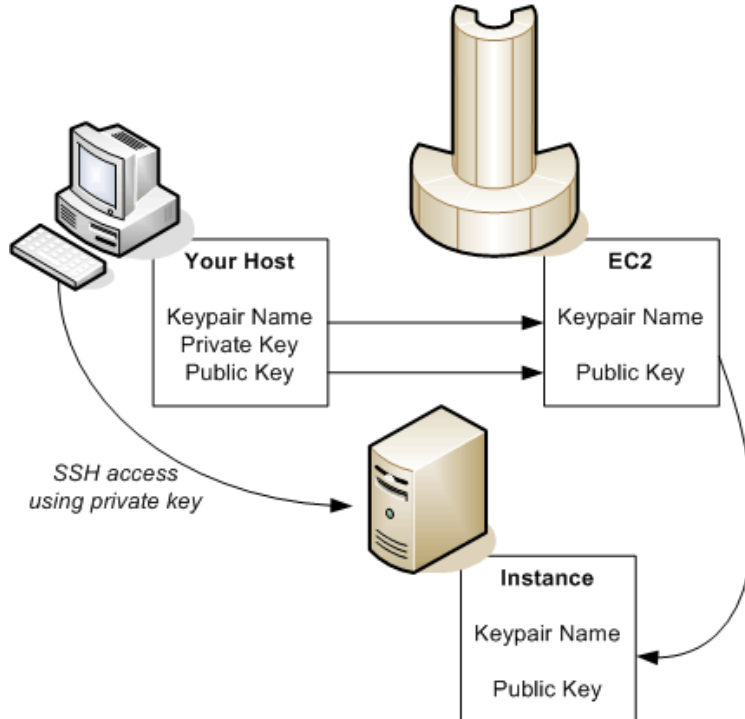
IMAGE    ami-e6cd298f    Key2047/test.manifest.xml    720208972571
available    private    x86_64    machine    windows
```

The response includes the image ID, the location of the file in Amazon S3, the image type, and whether the file is available.

2. Choose an AMI from the list and write down its AMI ID.

How to Generate a Key Pair

You must create a public/private key pair to ensure that only you have access to instances that you launch. After you generate a key pair, the public key is stored in Amazon EC2 using the key pair name you selected. Whenever you launch an instance using the key pair name, the public key is copied to the instance metadata. This allows you to access the instance securely using your private key.



To create a public/private key pair

1. Enter the following command:

```
PROMPT> ec2-add-keypair <keypair-name>
```

The `<keypair-name>` is the name you select for the key pair.

The resulting private key is displayed.

2. Open a text editor.
3. Paste the entire private key, starting with the line "`-----BEGIN RSA PRIVATE KEY-----`" and ending with the line "`-----END RSA PRIVATE KEY-----`".
4. Save the file and exit.



Note

This file should only be readable by the file owner.

Example

```
PROMPT> ec2-add-keypair gsg-keypair
KEYPAIR gsg-keypair
 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f
-----BEGIN RSA PRIVATE KEY-----
MIIEoQIBAAKCAQBULFg5ujHrtmljnutSuoO8Xe56LlT+HM8v/xkaa39EstM3/aFXTHgElQiJLChp
HungXQ29VTc8rc1bW0lkdi23OH5eqkMHGhvEwqa0HWASUMl14o3o/IX+0f2UcPoKCOVUR+jx71Sg
5AU52EQfanIn3ZQ8lFW7Edp5a3q4DhjG1UKT0HVbicL5E+g45zfb95wIyywWZfEW/UUF3LpGZyq/
ebIU1q1qTbHkLbCC2r7RTn8vpQWp47BGVYgtGSBMPTRP5hnbz zuqj3itkiLHjU39S2sJcJ0TrJx5
i8BygR4s3mHKBj8l+ePQxG1kGbF6R4yg6sECmXn17MRQVXODNHZbAgMBAAECggEAY1tsiUsIwD15
91CXirkYGuVfLyLflXenxfI50mDFms/mumTqloHO7tr0oriHDR5K7wMcY/YY5YkcXNo7mvUVD1pM
ZNUJs7rw9gZRTrf7LylaJ58kOcyajw8TsC4e4LPbFaHwSlD6K8rXh64o6WgW4SrsB6ICmrlkGQI7
3wcfgt5ecIu4TzF00E9IHjn+2eRlrsjBdeORI7KiUNC/pAG23I6MdDOFEQRcCSigCj+4/mciFUSA
SWS4dMbrpb9FNsIcf9dcLxVM7/6KxgJNfz9cXWzUw77Jg8x92Zd0fVhHOux5IZC+UvSKWB4dyfcI
tE8C3p9bbU9VGyY5vLCAiIb4qQKBgQDLiO24GXrIkswF32YtBBMuVgLGcWU9h9HlO9mKAc2m8Cm1
jUE5IpzRjTedc9I2qiIMUTwtgnw42auSCzbUeYMURPtDqyQ7p6AjMuJp9EPemcSVOK9vXYL0Ptco
xW9MC0dtV6iPkCN7gOqiZXPkRkaFbWADp16p8UAIvS/a5XXk5jwKBgQCKkPHi2EISh1uRkx1jyWC
iDCiK6JBRsMvpLbc0v5dKwP5a1o1fmdR5PJaV2qvZSj5CYNpMay1/EDNTY5OSIJU+0KfMQbyhsbm
rdLNLDL4+TcnT7c62/aH01ohYaf/VCbRhtLlBfGqQc7+sAc8vmKkesnF7CqCEKdyF/dhrxYdQKB
gC0iZzzNAapayz1+JcVTwwEid6j9JqNXbBc+Z2YwMi+T0Fv/P/hwkX/ypeOXniUcw0Ih/YtGBVAC
DQbsz7LcY1HqXiHKYNWNVXgww+oiChjxvEkSdsTTIfnK4VSCvU9BxDQhjdINDJbL6oar92UN7V
rBYvChJZF7LvUH4YmVpHAoGAbZ2X7XvoeEO+uZ58/BGK0IGHBYHBDiXtzmhdJr15HTYjxK7OgTZm
gK+8zp4L9IbVlGDMJO8vft32XPEWuvI8twCzFH+CsWLQADZMZKSSBasOZ/h1FwhdMgCMcY+Q1zd4
JZKjTSu3i7vhvx6RzdSedXEMNTZWN4q1Ix3kR5aHcukCgYA9T+ZrvmlF0seQPbLknn7EqhXIjBaT
P8TTvW/6bdPi23ExzxZn7K0drfclYRph1LHMPAONv/x2xALIf91UB+v5ohy1oDoasL0gij1houRe
2ERKKdwz0ZL9SWq6VTdhr/5G994CK72fy5WhyERbdjUIdHaK3M849JJuf8cSrvSb4g==
-----END RSA PRIVATE KEY-----
```

How to Launch the Instance

You are now ready to launch an instance of the AMI that you previously selected.

To launch an instance

1. Start the launch by entering the following command:

```
PROMPT> ec2-run-instances <ami_id> -k <keypair-name>
```

The `<ami_id>` is the AMI ID you selected earlier and `<keypair-name>` is the name of the key pair. The command will return the AMI instance ID, a unique identifier for each launched instance. You use the instance ID to manipulate the instance. This includes viewing the status of the instance, terminating the instance, and so on.

Launching the instance will take a few minutes.

2. View the progress of the instance by entering the following command:

```
PROMPT> ec2-describe-instances <instance_id>
```

The `<instance_id>` is the ID of the instance.

When the status field displays `running`, the instance was created and is booting.

3. Verify the instance is ready by entering the following command:

```
PROMPT> ec2-get-console-output <instance_id>
```

Look for the message: "Windows is Ready to use."



Note

The instance might not be immediately accessible over the network. Make sure to use the appropriate DNS name provided by the `ec2-describe-instances` command.



Important

Once you launch an instance, AWS bills you for all usage, including hourly CPU time. Make sure to terminate any instances that you do not want to leave running. For information on Amazon EC2 pricing, go to the [Amazon EC2 home page](#).

Example

The following example launches an instance of `ami-2bb65342`.

```
PROMPT> ec2-run-instances ami-2bb65342 -k gsg-keypair
RESERVATION    r-302dc059      416161254515    default
INSTANCE      i-eb977f82      ami-2bb65342    pending gsg-
keypair       0      m1.small    2007-10-16T07:56:20+0000    us-east-1a
```

The following shows the status of the launch:

```
PROMPT> ec2-describe-instances i-eb977f82
RESERVATION    r-302dc059      416161254515    default
INSTANCE      i-eb977f82      ami-2bb65342
ec2-72-44-40-222.compute-1.amazonaws.com    10-251-50-83.ec2.internal
running gsg-keypair    0      m1.small    2007-10-16T07:56:20+0000    us-
east-1a    windows
```

How to Get the Administrator Password

After you launch an instance, get its administrator password.



Note

If you launch a public AMI, you must get its administrator password. A rebundled AMI uses the last password that you set before bundling it.

Before you rebundle an AMI, you can change its administrator password. The new password will be the administrator password for all instances launched from this AMI.

To get the administrator password

- Enter the following command:

```
PROMPT> ec2-get-password -k gsg-keypair instance_id
```

The parameter `gsg-keypair` is the name of the file where you saved the private portion of the key pair you created and `instance_id` is the ID of the instance.

Amazon EC2 returns the Windows password.

Example

The following example gets the Windows password for instance `i-eb977f82`.

```
PROMPT> ec2-get-password -k id_rsa-gsg-keypair i-eb977f82
Qr89fdSlw
```

How to Authorize Network Access

To reach a running instance from the Internet, you must enable access for Remote Desktop on port 3389.

To enable Remote Desktop on port 3389

1. Get the public IP address of your local machine by going to a search engine, entering "what is my IP address," and using one of the provided services.
2. Authorize the security group to allow Remote Desktop access:

```
PROMPT> ec2-authorize default -p 3389 -s your_ip_address/32
PERMISSION    default  ALLOWS  tcp      3389      3389      FROM      CIDR
              your_ip_address/32
```

How to Connect to the Instance

After an instance starts, you can log in and modify it according to your requirements.

To connect to your instance

1. Retrieve the FQDN of your instance.
This example retrieves the FQDN of the `i-ae0bf0c7` instance.

```
PROMPT> ec2-describe-instances i-ae0bf0c7
RESERVATION  r-7430c31d  924417782495  default
INSTANCE     i-ae0bf0c7  ami-2bb65342
ec2-67-202-7-236.compute-1.amazonaws.com  ip-10-251-31-162.ec2.internal
running     gsg-keypair  0              m1.small
2008-03-21T16:19:25+0000  us-east-1a
```

In this example, the FQDN is `ec2-67-202-7-236.compute-1.amazonaws.com`

2. From the **Start** menu, point to **Programs**, point to **Accessories**, point to **Communications**, and click **Remote Desktop Connection**.
The Remote Desktop Connection dialog box appears.
3. Enter the FQDN in the **Computer** field and click **Connect**.
The Remote Desktop Connection client connects to the instance.
4. Enter `administrator` as the user name and enter the password you retrieved in [How to Get the Administrator Password \(p. 42\)](#).

You now have complete control over the instance. You can add, remove, modify, or upgrade packages and files to suit your needs.



Important

We recommend you exercise extreme care if you change any basic Amazon EC2 configuration settings. Otherwise, the AMI might become unbootable or inaccessible from the network once it is running.

How to Load Software and Make Changes

Now that you are logged into the Windows instance, you can load software and make changes as you would with any Windows server. When you are finished with your changes, you can bundle the changes as a new AMI and launch an identical copy at any time. For information on bundling AMIs, see [Bundling a Windows AMI \(p. 47\)](#)



Note

By default, Amazon EC2 instances running Windows do not have **Automatic Updates** enabled.

Bundling an AMI

Topics

- [Bundling a Linux or UNIX AMI \(p. 44\)](#)
- [Bundling a Windows AMI \(p. 47\)](#)

This section describes how to bundle an AMI for use with Amazon EC2.



Note

During bundling, only the root store is bundled. Data on other instance stores is not preserved.

Bundling a Linux or UNIX AMI

The AMI tools include three command line utilities:

- `ec2-bundle-image` bundles an existing AMI
- `ec2-bundle-volume` creates an AMI from an existing machine or installed volume
- `ec2-upload-bundle` uploads a bundled AMI to Amazon S3 storage

How to Install the AMI Tools

The AMI tools are available in both a zip file and as an RPM suitable for running on Fedora Core with Ruby 1.8.2 (or greater) installed. You need root privileges to install the software.

The AMI tools RPM is available from our [public Amazon S3 downloads bucket](#). For information about installing tools, refer to their provided documentation.

To install the AMI tools

1. Install Ruby using the yum package manager.

```
# yum install ruby
```

2. Install the AMI tools RPM.

```
# rpm -i ec2-ami-tools-x.x-xxxx.i386.rpm
```

Installation Issues

The AMI tools libraries install in `/usr/lib/site_ruby`.

If you receive a load error when running one of the AMI utilities, Ruby might not have found the path. To fix this, add `/usr/lib/site_ruby` to Ruby's library path, which is set in the `RUBYLIB` environment variable.

How to View Documentation

This section describes how to view Linux/UNIX documentation.

To view the manual for each utility

- Append `--manual` to the command that invokes the utility.

```
# ec2-bundle-image --manual
```

To view help for each utility

- Append `--help` to the command that invokes the utility.

```
# ec2-bundle-image --help
```

How to Bundle an AMI Using the AMI Tools

After creating a machine image, it must be bundled as an AMI for use with Amazon EC2. How you bundle the image depends on how you created the image (for information about creating AMIs, see [Creating an AMI \(p. 28\)](#)).

To bundle the loopback file image

- Enter the following command:

```
# ec2-bundle-image -i <image_name>.img -k <private_keyfile> -  
c <certificate_file> -u <user_id>
```

The `<image_name>` is the name of the image file, `<private_keyfile>` is the file that contains the private key, `<certificate_file>` is the file that contains the certificate, and `<user_id>` is the user ID associated with your account.



Note

The user ID is your AWS account ID without dashes. It is the same as your Amazon Access ID and consists of 12 digits.

To bundle a snapshot image (requires root privileges)

- Enter the following command:

```
# ec2-bundle-vol -k <private_keyfile> -c <certificate_file> -u <user_id>
```

The `<private_keyfile>` is the file that contains the private key, `<certificate_file>` is the file that contains the certificate, and `<user_id>` is the user ID associated with your account.



Note

Make sure to disable SELinux when running `ec2-bundle-vol`.



Note

The user ID is your AWS account ID without dashes. It is the same as your Amazon Access ID and consists of 12 digits.

Example

This command bundles an image created in a loopback file.

```
# ec2-bundle-image -i my-image.fs -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem
-c cert-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem -u AIDADH4IGTRXXKCD
image.part.00
image.part.01
...
image.part.NN
image.manifest.xml
```

This command bundles the local machine root file system.

```
# ec2-bundle-vol -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem -c cert-
HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem -u AIDADH4IGTRXXKCD
image.part.00
image.part.01
...
image.part.NN
image.manifest.xml
```

How to Upload a Bundled AMI

You must upload the bundled AMI to Amazon S3 before it can be accessed by Amazon EC2. Use `ec2-upload-bundle` to upload the bundled AMI that you created earlier. Amazon S3 stores data objects in buckets, which are similar to directories.

Buckets must have globally unique names. The `ec2-upload-bundle` utility uploads the bundled AMI to a specified bucket. If the specified bucket does not exist, it will be created. If the specified bucket exists and belongs to another user, the `ec2-upload-bundle` command will fail.

To upload the bundled AMI

- Enter the following command:

```
# ec2-upload-bundle -b <bucket> -m image.manifest.xml -a <access_key> -
s <secret_key>
```

The `<bucket>` is the target bucket, `<access_key>` is your AWS Access Key, and `<secret_key>` is your AWS Secret Key.

The AMI manifest file and all image parts are uploaded to Amazon S3. The manifest file is encrypted with the Amazon EC2 public key before being uploaded.

How to Register the AMI

You must register your image with Amazon EC2, so we can locate it and run instances based on it.



Note

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

To register the AMI that you created and uploaded to Amazon S3

- Execute the following command:

```
PROMPT> ec2-register <your-s3-bucket>/sampleimage.manifest.xml  
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.

Bundling a Windows AMI

This section describes how to bundle an AMI in Windows. The bundling process does the following:

- 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



Note

Before bundling an instance, you can configure the instance using the EC2Config service. For more information, see [Windows Configuration Service \(p. 138\)](#)

How to Bundle an AMI

Bundling your own AMIs allows you to make the most of Amazon EC2. Your AMIs become the basic unit of deployment which allow you to rapidly boot new custom instances as you need them.

All AMIs are loaded from Amazon S3 storage. You must upload the AMI to an existing account on Amazon S3.

Amazon S3 stores data objects in buckets, which are similar in concept to directories. You will need to specify a bucket name in the following example as `<your-s3-bucket>`. Buckets have globally unique names and are owned by unique users. 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. The **ec2-bundle-instance** utility uploads the bundled AMI to a specified bucket. If the specified bucket does not exist, it creates it. If the specified bucket belongs to another user, **ec2-bundle-instance** fails, and you have to try a different name.

For this, you will need your AWS Access Key ID (`<aws-access-key-id>`) and AWS Secret Access Key (`<aws-secret-access-key>`).

To bundle an AMI

1. Log in to the Windows instance and make any desired changes.



Note

We highly 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 of new instances using the `ec2-get-password` command.

- 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:\`.



Note

The `sdelete` utility is available from the [sdelete Download Page](#) or the [Microsoft Web Site](#).

- On the host where you have installed the API tools, enter the following command:

```
PROMPT> 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.

Amazon EC2 shuts down the instance, saves it as an AMI, and restarts it. You can launch copies of the AMI at any time in the future.

Example

```
PROMPT> ec2-bundle-instance i-eb977f82 -b mybucket -p myimage -
o AKIADQKE4SARGYLE -w eW91dHVizS5jb20vd2F0Y2g/dj1SU3NKMTlzeTNKSQ==
BUNDLE bun-e3a4418a i-eb977f82 mybucket myimage
2008-10-02T09:31:44+0000 2008-10-02T09:31:44+0000 pending
```

How to Monitor a Bundled AMI

Before you launch an AMI, you must wait for the bundling to complete and then register it. The bundling task moves from the "pending" state, to the "bundling" state, to the "storing" state, and finally to the "complete" state.

To view the status

- Enter the following command:

```
PROMPT> ec2-describe-bundle-tasks
```

Amazon EC2 returns output similar to the following:

```
BUNDLE bun-e3a4418a eb977f82 mybucket winami complete
2008-08-28T00:59:13+0000 2008-08-28T01:34:30+0000
```

How to Register the AMI

You must register your image with Amazon EC2, so we can locate it and run instances based on it.



Note

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

To register the AMI that Amazon EC2 created and uploaded to Amazon S3

- Execute the following command:

```
PROMPT> ec2-register <your-s3-bucket>/sampleimage.manifest.xml
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.

How to Share AMIs

Topics

- [Protecting a Shared AMI \(Linux and UNIX\) \(p. 49\)](#)
- [Sharing AMIs \(p. 53\)](#)

This section describes how to build and share AMIs.

Shared AMIs are AMIs that developers build and make available for other AWS developers to use. Building safe, secure, useable AMIs for public consumption is a fairly straightforward process, if you follow a few simple guidelines.

For information on building shared AMIs, see [Protecting a Shared AMI \(Linux and UNIX\) \(p. 49\)](#). For information on sharing AMIs, see [Sharing AMIs \(p. 53\)](#)

Protecting a Shared AMI (Linux and UNIX)

These guidelines are not requirements and you are welcome to follow or ignore them. However, following these guidelines produces a better user experience, helps ensure your users' *instances* are secure, and can protect you.

To build a shared AMI, follow these guidelines:

Shared AMI Guidelines

1	How to Update the AMI Tools at Boot Time (p. 49)
2	Disable Password-Based Logins for Root (p. 50)
3	Install Public Key Credentials (p. 51)
4	How to Disable sshd DNS Checks (optional) (p. 52)
5	Identify Yourself (p. 52)
6	Protect Yourself (p. 52)
7	Protect Paid AMIs (p. 53)



Note

These guidelines are written for Fedora distributions, but the principles apply to any AMI. You might need to modify the provided examples for other distributions. For other distributions, review their documentation or search the [AWS forums](#) in case someone else has done it already.

How to Update the AMI Tools at Boot Time

We recommend that your AMIs download and upgrade the Amazon EC2 AMI creation tools during startup. This ensures that new AMIs based on your shared AMIs will have the latest AMI tools.

To update the AMI tools at startup on Fedora

- Add the following to `rc.local`:

```
# Update the Amazon EC2 AMI creation tools
echo " + Updating ec2-ami-tools"
wget http://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.noarch.rpm && \
rpm -Uvh ec2-ami-tools.noarch.rpm && \
echo " + Updated ec2-ami-tools"
```

Use this method to automatically update other software on your image.



Note

When deciding which software to automatically update, consider the amount of WAN traffic that the update will generate (your users will be charged for it) and the risk of the update breaking other software on the AMI.



Note

The preceding procedure applies to Fedora distributions. For other distributions:

- On most Red Hat systems, add these steps to your `/etc/rc.d/rc.local` script.
- On Gentoo systems, add them to `/etc/conf.d/local.local`.
- On Ubuntu systems, add them to `/etc/rc.local`.
- On Debian, you might need to create a start up script in `/etc/init.d` and use `update-rc.d <scriptname> defaults 99` (where `<scriptname>` is the name of the script you created) and add the steps to this script.

Disable Password-Based Logins for Root

Using a fixed root password for a public AMI is a security risk that can quickly become known. Even relying on users to change the password after the first login opens a small window of opportunity for potential abuse.

To solve this problem, disable password-based logins for the root user. Additionally, we recommend you randomize the root password at boot.

To disable password-based logins for root

1. Open the `/etc/ssh/sshd_config` file with a text editor and locate the following line:

```
#PermitRootLogin yes
```

2. Change the line to:

```
PermitRootLogin without-password
```

The location of this configuration file might differ for your distribution, or if you are not running OpenSSH. If this is the case, consult the relevant documentation.

3. To randomize the root password, add the following to your boot process:

```
if [ -f "/root/firstrun" ] ; then
  dd if=/dev/urandom count=50|md5sum|passwd --stdin root
  rm -f /root/firstrun
else
  echo "* Firstrun *" && touch /root/firstrun
```

```
fi
```



Note

This step assumes that a `/root/firstboot` file is bundled with the image. If file was not created, the root password will never be randomized and will be set to the default.



Note

If you are using a distribution other than Fedora, you might need to consult the documentation that accompanied the distribution.

Remove SSH Host Key Pairs

If you plan to share an AMI derived from a public AMI, remove the existing SSH host key pairs located in `/etc/ssh`. This forces SSH to generate new unique SSH key pairs when someone launches an instance using your AMI, improving security and reducing the likelihood of "man-in-the-middle" attacks.

The following list shows the SSH files to remove.

- `ssh_host_dsa_key`
- `ssh_host_dsa_key.pub`
- `ssh_host_key`
- `ssh_host_key.pub`
- `ssh_host_rsa_key`
- `ssh_host_rsa_key.pub`

Install Public Key Credentials

After configuring the AMI to prevent logging in using a password, you must make sure users can log in using another mechanism.

Amazon EC2 allows users to specify a public-private key pair name when launching an instance. When a valid key pair name is provided to the `RunInstances` API call (or through the command line API tools), the public key (the portion of the key pair that Amazon EC2 retains on the server after a call to `CreateKeyPair`) is made available to the instance through an HTTP query against the instance metadata.

To login through SSH, your AMI must retrieve the key value at boot and append it to `/root/.ssh/authorized_keys` (or the equivalent for any other user account on the AMI). Users will be able to launch instances of your AMI with a key pair and log in without requiring a root password.

```
if [ ! -d /root/.ssh ] ; then
    mkdir -p /root/.ssh
    chmod 700 /root/.ssh
fi
# Fetch public key using HTTP
curl http://169.254.169.254/2009-04-04//meta-data/public-keys/0/openssh-key
  > /tmp/my-key
if [ $? -eq 0 ] ; then
    cat /tmp/my-key >> /root/.ssh/authorized_keys
    chmod 700 /root/.ssh/authorized_keys
    rm /tmp/my-key
fi
```

This can be applied to any user account; you do not need to restrict it to root.



Note

Rebundling an instance based on this image includes the key with which it was launched. To prevent the key's inclusion, you must clear out (or delete) the `authorized_keys` file or exclude this file from rebundling.

How to Disable sshd DNS Checks (optional)

Disabling sshd DNS checks slightly weakens your sshd security. However, if DNS resolution fails, SSH logins will still work. If you do not disable sshd checks, DNS resolution failures prevent all logins.

To disable sshd DNS checks

1. Open the `/etc/ssh/sshd_config` file with a text editor and locate the following line:

```
#UseDNS yes
```

2. Change the line to:

```
UseDNS no
```



Note

The location of this configuration file can differ for your distribution or if you are not running OpenSSH. If this is the case, consult the relevant documentation.

Identify Yourself

Currently, there is no easy way to know who provided a shared AMI as each AMI is represented by a numeric user ID.

We recommend that you post a description of your AMI, and the AMI ID, in the Amazon EC2 developer forum. This provides a convenient central location for users who are interested in trying new shared AMIs.

Protect Yourself

The previous sections described how to make your shared AMIs safe, secure, and useable for the users who launch them. This section describes guidelines to protect yourself from the users of your AMI.

We recommend against storing sensitive data or software on any AMI that you share. Users who launch a shared AMI might be able to rebundle it and register it as their own. Follow these guidelines to help you to avoid some easily overlooked security risks:

- Always delete the shell history before bundling. If you attempt more than one bundle upload in the same image, the shell history contains your secret access key.
- Bundling a running instance requires your private key and X.509 certificate. Put these and other credentials in a location that is not bundled (such as the instance store).
- Exclude the ssh authorized keys when bundling the image. The Amazon public images store the public key used to launch an instance with its ssh authorized keys file.



Note

Unfortunately, it is not possible for this list of guidelines to be exhaustive. Build your shared AMIs carefully and take time to consider where you might expose sensitive data.

Protect Paid AMIs

The simplest way to prevent users from rebundling Paid AMIs that you create is to not provide root access to the AMI and to pay attention to security announcements that involve privilege escalations. Amazon EC2 requires you to have root access any AMI that you rebundle.

If you must provide root access to an AMI, Amazon EC2 tools are designed to protect the product code. Although this is effective, it is not guaranteed and users might create AMIs using other tools.

To ensure users cannot rebundle your paid AMIs, we recommend that you configure your application to check the instance metadata to verify that the product code is intact.

Sharing AMIs

Amazon EC2 enables users to share their AMIs with other users. This section describes how to share AMIs using the Amazon EC2 command line tools.



Note

Before proceeding, make sure to read the security considerations of sharing AMIs in the [Protecting a Shared AMI \(Linux and UNIX\) \(p. 49\)](#) section.

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

The `launchPermission` attribute is a list of users and *launch groups*. *Launch permissions* can be granted by adding or removing items from the list. Explicit launch permissions for users are granted or revoked by adding or removing their 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 users. Users launching the AMI are billed.

Select from the following:

- [How to Make an AMI Public \(p. 53\)](#)
- [How to Share an AMI with Specific Users \(p. 54\)](#)
- [How to Publish Shared AMIs \(p. 55\)](#)

How to Make an AMI Public

To make an AMI public

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

```
PROMPT> ec2-modify-image-attribute <ami_id> --launch-permission -a all
```

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

To check the launch permissions of an AMI

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

```
PROMPT> ec2-describe-image-attribute <ami_id> -l
```

To make an AMI private again

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

```
PROMPT> 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.

Example

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

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

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

```
PROMPT> ec2-describe-image-attribute ami-2bb65342 -l
launchPermission      ami-2bb65342      group      all
```

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

```
PROMPT> ec2-modify-image-attribute ami-2bb65342 -l -r all
launchPermission      ami-2bb65342      REMOVE    group      all
```

How to Share an AMI with Specific Users

You can share an AMI with specific users without making the AMI public. All you need is the user's AWS user's account ID, which is available on the AWS Account Activity page.

To grant explicit launch permissions

- Enter the following command:

```
PROMPT> ec2-modify-image-attribute <ami_id> -l -a <user_id>
```

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

To remove launch permissions for a user

- Enter the following command:

```
PROMPT> ec2-modify-image-attribute <ami_id> -l -r <user_id>
```

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

To remove all launch permissions

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

```
PROMPT> ec2-reset-image-attribute <ami_id> -l
```

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



Note

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

Example

The following example grants launch permissions to the AIDADH4IGTRXXKCD user for the ami-2bb65342 AMI:

```
PROMPT> ec2-modify-image-attribute ami-2bb65342 -l -a AIDADH4IGTRXXKCD
launchPermission      ami-2bb65342      ADD      userId  AIDADH4IGTRXXKCD
```

The following example removes launch permissions from the AIDADH4IGTRXXKCD user for the ami-2bb65342 AMI:

```
PROMPT> ec2-modify-image-attribute ami-2bb65342 -l -r AIDADH4IGTRXXKCD
launchPermission      ami-2bb65342      REMOVE  userId  AIDADH4IGTRXXKCD
```

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

```
PROMPT> ec2-reset-image-attribute ami-2bb65342 -l
launchPermission      ami-2bb65342      RESET
```

How to Publish Shared AMIs

After creating a shared AMI, other developers can find it in the [Amazon EC2 Resource Center](#).

To publish your AMI

1. Post it in the Public AMIs Folder of the [Amazon Web Services Resource Center](#), including the following information:
 - AMI ID
 - AMI manifest
2. Also, add the following information (recommended, but not required):
 - Publisher
 - Publisher URL
 - OS / Distribution
 - Key Features
 - Description
 - Daemons / Services
 - Release Notes
3. If you want to, you can paste the following information into the document. You must be in HTML edit mode.

```
<strong>AMI ID: </strong>[ami-id]<br />
<strong>AMI Manifest: </strong>[bucket/image.manifest.xml]<br />
<h2>About this &AMI;</h2>
<ul>

    <li>Published by [Publisher] (<a href="http://www.mysite.com">[http://
www.mysite.com]</a>).<br />
    </li>
    <li>[Key Features] <br />
    </li>
    <li>[Description]</li>
```


- Amazon Payments handles payment processing.



Basic DevPay Flow

1	Your customer uses an Amazon.com account to sign up and pay for your AMI. The sign-up page indicates that you have teamed up with Amazon Payments to make billing easy and secure.
2	Your customer pays the price you've defined to use your product.
3	DevPay subtracts a fixed transaction fee and pays you the difference.
4	You pay the costs of Amazon EC2 that your AMI used, and a percentage-based DevPay fee.

For more information about Amazon DevPay, refer to the *Amazon DevPay Developer Guide*.

Summary of How Paid AMIs Work

With a paid AMI, your customers:

- Must be signed up to use Amazon EC2 themselves
- Buy your paid AMI and then launch instances of it
- Always use *their own* AWS credentials when launching instances; you don't launch instances of your paid AMI for them with your credentials
- Pay the price you set for the paid AMI, and not the normal Amazon EC2 rates



Important

The discounts you get with Amazon EC2 Reserved Instances don't apply to Amazon DevPay products. That is, if you purchase Reserved Instances, you don't get the lower price associated with them when your customers launch your paid or supported AMIs. Also, if your customers purchase Reserved Instances, when they use your paid or supported AMIs, they continue to pay the price you specified for the use of your paid or supported AMIs. For more information about Reserved Instances, see [Reserved Instance Concepts \(p. 14\)](#).

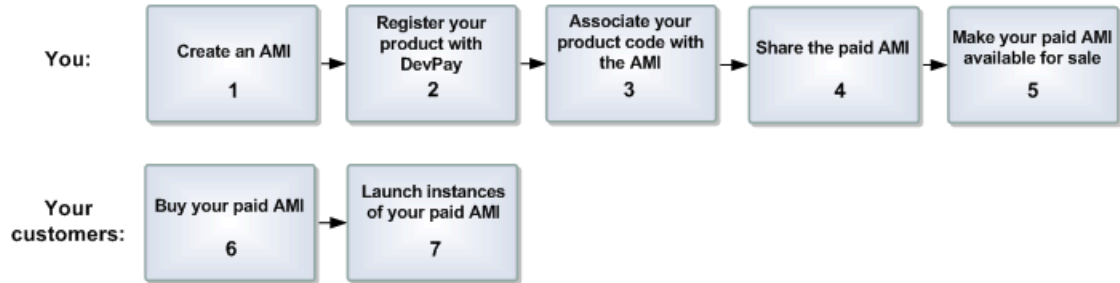
You can also use Amazon EC2 and Amazon DevPay together with a *supported AMI*. For more information about supported AMIs, see [Supported AMIs \(p. 62\)](#).

The following figure and table summarize the basic flow for creating and using paid AMIs.




Note

Detailed information about most of the following steps is provided in the *Amazon DevPay Developer Guide*.



Paid AMI Process

1	You create an AMI as described elsewhere in this guide.
2	You register a product with Amazon DevPay. For more information, see Product Registration (p. 59) . As part of this process, you provide a product description, product pricing, etc. This registration process creates a product code for the product and a URL where customers can sign up to use the product (called the <i>purchase URL</i>).
3	You use an Amazon EC2 command or API call to associate the product code with your AMI. For more information, see How to Associate a Product Code with an AMI (p. 60) . This makes the AMI a paid AMI.
4	You use an Amazon EC2 command or API call to share the AMI with select customers or the public. For more information, see How to Share Your Paid AMI (p. 61) .
	 Note Even if you share a paid AMI and it has a product code, no one can use the AMI until they sign up for it (see the following steps).
5	You make your paid AMI available for sale. To do this, you make the aforementioned purchase URL available. You can advertise your paid AMI in the Solutions Catalog on the AWS Developer Connection site and on the Amazon Machine Images (AMIs) page on the AWS Resource Center.
6	Customers use the purchase URL you provide to sign up for and purchase your product. If they're not already signed up for Amazon EC2, they'll be prompted to sign up. They purchase your product with their Amazon.com accounts. They must have the credentials needed to launch Amazon EC2 instances. At this point, they have the AMI ID (from step 5).
7	Customers then launch an Amazon EC2 instance specifying the AMI ID. Because you associated the shared AMI with the product code, the customers are charged at the rate you set. For more information, see Paying for AMIs (p. 78) .



Note

You can associate your DevPay product code with more than one AMI. However, a single AMI can be associated with only one product code. If you plan to sell multiple AMIs, you could sell them all under a single product code, or different product codes (by registering multiple DevPay products). For information about why you might choose a single product code

or multiple product codes, go to [If You Have Multiple AMIs to Sell](#) in the *Amazon DevPay Developer Guide*.

Each customer's bill for the AMI is displayed on their Application Billing page, which shows the activity for DevPay products. Also, at any time, you can confirm the customer is still currently subscribed to your product. For more information, refer to the *Amazon DevPay Developer Guide*.



Note

In the preceding process, you associate your product code with your own AMI and sell the AMI as a DevPay product. There's another scenario for using DevPay with Amazon EC2 in which you sell software or a service to EC2 users and let them associate your product code with their own AMIs. For more information, see [Supported AMIs \(p. 62\)](#).

The Product Code and AMI Rebundling

Associating a product code with an AMI turns it into a paid AMI that EC2 users must sign up for to use. Can you ensure that the product code stays with the AMI if someone rebundles the AMI? The answer varies for Linux/UNIX AMIs and Windows AMIs. These are described in the following sections.

Linux/UNIX AMIs

If you give the customer root access to your paid Linux/UNIX AMI, the customer can rebundle it (for more information, see [Bundling a Linux or UNIX AMI \(p. 44\)](#)). If your customer uses AWS tools to rebundle the AMI, the rebundled AMI inherits the product code. When launching instances of the rebundled AMI, the customer is still billed for usage based on your price. However, if the customer doesn't use the AWS tools when rebundling, the rebundled AMI won't inherit the product code, and the customer will pay normal Amazon EC2 rates and not your price. Also, a customer with root access could find some other way to remove the product code from the AMI.

When a customer contacts you for support for a paid AMI, you can confirm your product code is associated with the AMI and the customer's instance is currently running the AMI. For more information, go to [How to Confirm an Instance Is Running with a Product Code \(p. 61\)](#).

If you have software installed on the AMI, the software can retrieve the instance metadata to determine if the product code is associated with the instance. For more information, see [How to Get the Product Code from Within an Instance \(p. 62\)](#).

Keep in mind that the preceding methods for confirming the association of the product code with the instance are not foolproof because a customer with root access to the instance could return false information indicating the product code is associated with the instance.

Windows AMIs

When you associate a product code with a Windows AMI, the association is permanent. Therefore, we recommend you keep a separate, base copy of the AMI that has no product code associated with it.

Anyone who purchases a Windows AMI can rebundle it (for more information, see [Bundling a Windows AMI \(p. 47\)](#)). The product code is automatically transferred to the rebundled AMI. When EC2 users launch instances of the rebundled AMI, they pay the rates you set when you registered your DevPay product. In turn, you're charged for the EC2 costs they incur.

Product Registration

You must register a product with Amazon DevPay. The product can cover a single AMI that you want to sell or multiple AMIs. During registration, you provide product information such as pricing, and you receive information you need to sell your product.



Important

The *Amazon DevPay Developer Guide* covers the procedure for registering your product with Amazon DevPay. Before you register your product, we recommend you read the information in that guide about how to set your AMI's price and how billing for Amazon DevPay products works.

You provide the following information during registration:

- Company name
- Product name
- Product description (as you want your customers to see it)
- Redirect URL (the page you want customers to see after they have purchased the product)
- Any terms and conditions you want displayed (optional)
- Contact e-mail address and telephone number (to be used by AWS and not displayed to customers)
- Contact e-mail or URL (to be displayed to customers)
- The specific regions, environments, and instance types the product covers
- Pricing for use of the product (you can set different prices based on region, environment, and instance type)

The information you display at the redirect URL should give information about the AMI.

Registration provides you with the following information:

- Product code
- Product token
- Purchase URL

You need the product code and purchase URL to integrate your product with DevPay as described in [Summary of How Paid AMIs Work \(p. 57\)](#) and [Supported AMIs \(p. 62\)](#). You need the product token if you're going to set up your system to later verify whether a customer is still subscribed to your product. For more information, refer to the *Amazon DevPay Developer Guide*.



Note

AWS must approve your product after you register it. The approval process typically takes one business day.

How to Associate a Product Code with an AMI

You must be the owner of an AMI to associate a product code with it. Each AMI can have only a single product code associated with it, but you can associate a single product code with more than one AMI. You might do this if you have similar versions of an AMI (for example, a 32-bit version and a 64-bit version), you've assigned them all the same price, and you'd like to minimize the number of Amazon DevPay product codes you have (to make your bookkeeping easier).

To associate a product code with an AMI

- Enter the following command:

```
PROMPT> ec2-modify-image-attribute <ami_id> --product-code <product_code>
```

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

To verify the product code is associated with the AMI

- Enter the following command:

```
PROMPT> ec2-describe-image-attribute <ami_id> --product-code
```

You can't change or remove the `productCodes` attribute after you've set it. If you want to use the same image without the product code or associate a different product code with the image, you must reregister the image to obtain a new AMI ID. You can then use that AMI without a product code or associate the new product code with the AMI ID.

Example

The following example associates the `ami-2bb65342` AMI with the `774F4FF8` product code.

```
PROMPT> ec2-modify-image-attribute ami-2bb65342 --product-code 774F4FF8
```

```
productCodes      ami-2bb65342          productCode      774F4FF8
```

This example verifies that the product code is associated with the AMI.

```
PROMPT> ec2-describe-image-attribute ami-2bb65342 --product-code
```

```
productCodes      ami-2bb65342          productCode      774F4FF8
```

How to Share Your Paid AMI

After you associate the product code with the AMI, you need to share the AMI with select customers or the public by using the `ec2-modify-image-attribute` command.

To share the AMI

- Enter the following command:

```
PROMPT> ec2-modify-image-attribute <ami_id> --launch-permission -a all
```

The `<ami_id>` is the AMI ID.

Even though you've shared the AMI, no one can use it until they sign up for your product by going to the purchase URL. Once customers sign up, any instances of the paid AMI they launch will be billed at the rate you specified during product registration.

Example

The following example shares the `ami-2bb65342` AMI with the public.

```
PROMPT> ec2-modify-image-attribute ami-2bb65342 --launch-permission -a all
```

```
launchPermission  ami-2bb65342  ADD  group  all
```

How to Confirm an Instance Is Running with a Product Code

If you have created a product for others to use with their AMIs (the supported AMI scenario), you might want to confirm that a particular AMI is associated with your product code and a particular instance is currently running that AMI.



Note

You must be the owner of the product code to successfully call **ec2-confirm-product-instance** with that product code.

Because your customers don't own the product code, they should describe their instances to confirm their instances are running with your product code.

To confirm an instance is running an AMI associated with your product code

- Enter the following command:

```
PROMPT> ec2-confirm-product-instance <product_code> -i <instance>
```

The `<product_code>` is the product code and `<instance>` is the instance.

If the AMI is associated with the product code, `true` is returned with the AMI owner's account ID. Otherwise, `false` is returned.

Example

The following example confirms whether the `i-10a64379` instance is running the `6883959E` product code.

```
PROMPT> ec2-confirm-product-instance 6883959E -i i-10a64379
```

```
6883959E i-10a64379 true 495219933132
```

How to Get the Product Code from Within an Instance

A running Amazon EC2 instance can determine if it has an Amazon DevPay product code. The instance retrieves the product code similarly to how it retrieves other metadata. For more information about retrieving metadata, see [Instance Metadata \(p. 71\)](#).

To retrieve a product code, query a web server with this REST-like API call:

```
GET http://169.254.169.254/2007-03-01/meta-data/product-codes
```

Amazon EC2 returns a response similar to the following:

```
774F4FF8
```

Supported AMIs

Supported AMIs are different from paid AMIs. With a supported AMI, you charge for software or a service you provide that customers use with their own AMIs.

The main difference between a [paid AMI](#) and a *supported AMI* is how the AMI is associated with a product code:

- **Paid AMI**—You associate your own product code with your own AMI
- **Supported AMI**—Other EC2 users associate your product code with their own AMIs

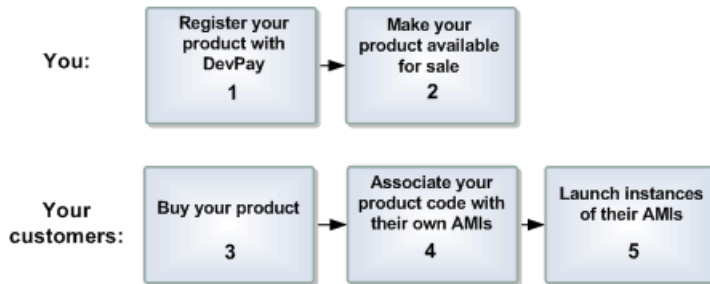


Important

If your customers purchase Reserved Instances, they don't get the Reserved Instance price discount with supported AMIs. That is, if they associate your product code with their AMIs, they don't get the lower price associated with their Reserved Instances when they launch

those AMIs. They always pay the price that you specified for your DevPay product. For more information about Reserved Instances, see [Reserved Instance Concepts \(p. 14\)](#).

The following figure and table summarizes the flow for creating and using supported AMIs.



Supported AMI Process

1	You register a product with Amazon DevPay. For more information, see Product Registration (p. 59) . As part of this process, you provide a product description, product pricing, etc. This registration process creates a product code for the product and a URL where customers can sign up to use the product (called the <i>purchase URL</i>).
2	You make your product available for sale.
3	Customers use the purchase URL to sign up for and purchase your product. If they're not already signed up for Amazon EC2, they'll be prompted to sign up. They purchase your product with their Amazon.com accounts. They must have the credentials needed to launch Amazon EC2 instances. At this point, they have the product code (from step 2).
4	Customers then use an Amazon EC2 command or API call to associate the product code with their AMIs. For more information, see How to Associate a Product Code with an AMI (p. 60) .
5	Customers then launch one or more instances of the AMIs. Because the customers associated their AMIs with the product code, they are charged at the rate you set.



Note

Amazon EC2 prevents your customers (but not you as the product code owner) from associating your product code with AMI types the product isn't configured for. For example, if the product is configured only for Linux/UNIX AMIs, your customers can't associate the product code with Windows AMIs. Also, Amazon EC2 prevents your customers from launching specific instance types your product isn't configured for. For more information about product configuration, go to [Your Product's Configuration and Price](#) in the *Amazon DevPay Developer Guide*.

Each customer's bill for the AMI is displayed on their Application Billing page, which shows the activity for DevPay products. For more information, refer to the *Amazon DevPay Developer Guide*.

When a customer contacts you for support for an AMI, you can confirm your product code is associated with the AMI and the customer's instance is currently running the AMI. For more information, see [How to Confirm an Instance Is Running with a Product Code \(p. 61\)](#).

Launching and Using Instances

Topics

- [How to Find a Suitable AMI \(p. 64\)](#)
- [How to Generate an SSH Key Pair \(p. 66\)](#)
- [How to Add Rules to the Default Security Group \(p. 68\)](#)
- [How to Run an Instance \(p. 69\)](#)
- [Instance Metadata \(p. 71\)](#)
- [Instance Storage \(p. 75\)](#)
- [Using Shared AMIs \(p. 76\)](#)
- [Paying for AMIs \(p. 78\)](#)
- [Getting Console Output and Rebooting Instances \(p. 81\)](#)
- [Related Topics \(p. 82\)](#)

This section describes how to launch *instances* and retrieve instance-specific data from within the instance. It also covers launching *shared AMIs* and security risks associated with running shared AMIs.



Note

If you create an instance in one region, you cannot launch it in another region without migrating it. For information on regions, see [Region and Availability Zone Concepts \(p. 19\)](#). For information on migrating AMIs, refer to the `ec2-migrate-bundle` section in the [Amazon Elastic Compute Cloud Command Line Reference](#).

How to Find a Suitable AMI

This section describes how to find an AMI.

AWS Management Console

To find a suitable AMI

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **AMIs** in the **Navigation** pane.
The console displays your AMIs and all public AMIs.
3. To reduce the number of displayed AMIs, select options from the **Viewing** list boxes. For example, you might want to display Amazon images.
4. After locating your desired AMI, write down its AMI ID. You can use this to launch instances of the AMI or register your own AMI, using this as a baseline.

Command Line Tools

To find a suitable AMI

1. Use the `ec2-describe-images` command.

```
PROMPT> ec2-describe-images -o self -o amazon | grep machine
```

```
IMAGE      ami-2c5fba45      ec2-public-images/demo-paid-AMI-v1.07.manifest.xml  
           amazon      available      public      A79EC0DB      i386      machine
```

Amazon Elastic Compute Cloud User Guide

How to Find a Suitable AMI

```
IMAGE    ami-bd9d78d4    ec2-public-images/demo-paid-AMI.manifest.xml
amazon   available      public   A79EC0DB   i386     machine
IMAGE    ami-2f5fba46    ec2-public-images/developer-image-i386-
v1.07.manifest.xml    amazon   available      public   i386     machine
IMAGE    ami-26b6534f    ec2-public-images/developer-image.manifest.xml
amazon   available      public   i386     machine
IMAGE    ami-f51aff9c    ec2-public-images/fedora-8-i386-base-
v1.06.manifest.xml    amazon   available      public   i386     machine
aki-a71cf9ce    ari-a51cf9cc
IMAGE    ami-2b5fba42    ec2-public-images/fedora-8-i386-base-
v1.07.manifest.xml    amazon   available      public   i386     machine
aki-a71cf9ce    ari-a51cf9cc
IMAGE    ami-f21aff9b    ec2-public-images/fedora-8-x86_64-base-
v1.06.manifest.xml    amazon   available      public   x86_64
machine    aki-b51cf9dcari-b31cf9da
IMAGE    ami-2a5fba43    ec2-public-images/fedora-8-x86_64-base-
v1.07.manifest.xml    amazon   available      public   x86_64
machine    aki-b51cf9dcari-b31cf9da
IMAGE    ami-a21affcb    ec2-public-images/fedora-core-6-x86_64-base-
v1.06.manifest.xml    amazon   available      public   x86_64
machine    aki-a53adfccari-a23adfcb
IMAGE    ami-2d5fba44    ec2-public-images/fedora-core-6-x86_64-base-
v1.07.manifest.xml    amazon   available      public   x86_64
machine    aki-a53adfccari-a23adfcb
IMAGE    ami-225fba4b    ec2-public-images/fedora-core4-apache-mysql-
v1.07.manifest.xml    amazon   available      public   i386     machine
IMAGE    ami-25b6534c    ec2-public-images/fedora-core4-apache-
mysql.manifest.xml    amazon   available      public   i386     machine
IMAGE    ami-2e5fba47    ec2-public-images/fedora-core4-apache-
v1.07.manifest.xml    amazon   available      public   i386     machine
IMAGE    ami-23b6534a    ec2-public-images/fedora-core4-apache.manifest.xml
amazon   available      public   i386     machine
IMAGE    ami-215fba48    ec2-public-images/fedora-core4-base-
v1.07.manifest.xml    amazon   available      public   i386     machine
IMAGE    ami-20b65349    ec2-public-images/fedora-core4-base.manifest.xml
amazon   available      public   i386     machine
IMAGE    ami-205fba49    ec2-public-images/fedora-core4-i386-base-
v1.07.manifest.xml    amazon   available      public   i386     machine
aki-9b00e5f2
IMAGE    ami-255fba4c    ec2-public-images/fedora-core4-mysql-
v1.07.manifest.xml    amazon   available      public   i386     machine
IMAGE    ami-22b6534b    ec2-public-images/fedora-core4-mysql.manifest.xml
amazon   available      public   i386     machine
IMAGE    ami-36ff1a5f    ec2-public-images/fedora-core6-base-
x86_64.manifest.xml    amazon   available      public   x86_64
machine
IMAGE    ami-235fba4a    ec2-public-images/getting-started-
v1.07.manifest.xml    amazon   available      public   i386     machine
IMAGE    ami-2bb65342    ec2-public-images/getting-started.manifest.xml
amazon   available      public   i386     machine
```

The command lists your AMIs and Amazon's public AMIs. The output might not exactly match the preceding example.

2. Look for the line containing the public image identified by the `ec2-public-images/getting-started.manifest.xml` value in the third column and note the corresponding value in the second column.

This is the AMI ID you need. In this example, it is `ami-2bb65342`.

How to Generate an SSH Key Pair

When you run an instance of a public AMI, it has no password and you need a public/private key pair to log in to the instance. One half of this key pair is embedded in your instance, allowing you to log in securely without a password using the other half of the key pair. After learning to create your own images, you can choose other mechanisms to allow you to securely login to your new instances. Every key pair you generate requires a name. Be sure to choose a name that is easy to remember.



Note

If you are using PuTTY in Windows, convert the private key to PuTTY's format. For more information on using PuTTY with Amazon EC2, go to the [Amazon Elastic Compute Cloud Getting Started Guide](#).

AWS Management Console

To generate a key pair

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Key Pairs** in the **Navigation** pane.
The console displays a list of key pairs associated with your account.
3. Click **Create Key Pair**.
The **Key Pair** dialog box appears.
4. Enter a name for the new key pair in the **Key Pair Name** field and click **Create**.
You are prompted to download the key file.
5. Download the key file and keep it in a safe place. You will need it to access any instances that you launch with this key pair.

Command Line Tools

To generate a key pair using `gsg-keypair`

1. Enter the following information.

```
PROMPT> ec2-add-keypair gsg-keypair
```

Amazon EC2 returns a key pair, similar to the key pair in the following example.

```
KEYPAIR gsg-keypair
 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f
-----BEGIN RSA PRIVATE KEY-----
MIIEoQIBAAKCAQBULFg5ujHrtmljnutSuoO8Xe56LlT+HM8v/xkaa39EstM3/
aFXTHgElQiJLChp
HungXQ29VTc8rc1bW0lkdi23OH5eqkMHGhvEwqa0HWASUM1l4o3o/IX+0f2UcPoKCOVUR
+jx7lSg
5AU52EQfanIn3ZQ8lFW7Edp5a3q4DhjG1UKToHVbicL5E+g45zfB95wIyywWZfeW/
UUF3LpGZyq/
ebIUlq1qTbHkLbCC2r7RTn8vpQWp47BGVYgtGSBMPTRP5hnbzZuqj3itkiLHjU39S2sJcJ0TrJx5
i8BygR4s3mHKBj8l
+ePQxG1kGbF6R4yg6sECmXn17MRQVXODNHZbAgMBAAEcggEAY1tsiUsIwDl5
91CXirkYGuVfLyLflXenxfI50mDFms/mumTqloHO7tr0oriHDR5K7wMcY/
YY5YkcXNo7mvUVD1pM
ZNUJs7rw9gZRTrf7LylaJ58kOcyajw8TsC4e4LPbFaHwSld6K8rXh64o6WgW4SrsB6ICmr1kGQI7
3wcfgt5ecIu4Tzf0OE9IHjn+2eRlrsjBdeORi7KiUNC/pAG23I6MdDOFEQRcCSigCj+4/
mciFUSA
```

Amazon Elastic Compute Cloud User Guide

How to Generate an SSH Key Pair

```
SWS4dMbrpb9FNsIcf9dcLxVM7/6KxgJNfZc9XWzUw77Jg8x92Zd0fVhHOux5IZC
+UvSKWB4dyfcI
tE8C3p9bbU9VgY5vLCAiIb4qQKBgQDLiO24GXrIkswF32YtBBMuVgLGcWu9h9H1O9mKAc2m8Cm1
jUE5IpzRjTedc9I2qiIMUTwtgnw42auSCzbUeYMURPtDqyQ7p6AjMuJp9EPemcSVOK9vXYL0Ptco
xW9MC0dtV6iPkCN7gOqiZXPRKaFbWADp16p8UAIvS/
a5XXk5jwKBgQCKkPHi2EIShluRkxh1jyWC
iDCiK6JBRsMvpLbc0v5dKwP5alo1fmdR5PJaV2qvZSj5CYNpMAy1/EDNTY5OSIJU
+0KFmQbyhsbm
rdLNLDL4+TcnT7c62/aH01ohYaf/VcBRhtLlBfqGoQc7+sAc8vmKkesnF7CqCEKDyF/
dhrxYdQKB
gC0iZzzNAapayz1+JcVTtwEid6j9JqNXbBc+Z2YwMi+T0Fv/P/hwkX/ypeOXnIUcw0Ih/
YtGBVAC
DQbsz7LcY1HqXiHKYNWNvXgww0
+oiChjxvEkSdsTTifnK4VScvU9BxDQhjdINDJbL6oar92UN7V
rBYvChJZF7LvUH4YmVpHAoGAbZ2X7XvoeEO+uZ58/
BGKOIGHByHBDiXtzMhdJr15HTYjxK7OgTZm
gK+8zp4L9IbvLGDMJO8vft32XPEWuvI8twCzFH+CsWLQADZMZKSsBasOZ/h1FwhdMgCmCY
+Q1zd4
JZKjTSu3i7vhvx6RzdSedXEMNTZWN4qlIx3kR5aHcukCgYA9T
+Zrvm1F0seQPbLknn7EqhXIjBaT
P8TTvW/6bdPi23ExzxZn7KODrfclYRph1LHMpAONv/x2xALIf91UB
+v5ohy1oDoasL0gijlhouRe
2ERKkdWz0ZL9SWq6VTdhr/5G994CK72fy5WhyERbdjUIIdHaK3M849JJuf8cSrvSb4g==
-----END RSA PRIVATE KEY-----
```

The private key returned must be saved to a local file so that you can use it later.

2. Create a file named `id_rsa-gsg-keypair` and paste the entire key generated in step 1, including the following lines.

```
"-----BEGIN RSA PRIVATE KEY-----"
"-----END RSA PRIVATE KEY-----"
```

3. Confirm that the file contents looks similar to the following and save the file.

You can save the file in any directory, but if you do not put it in your current directory, you should specify the full path when using commands that require the key pair.

```
-----BEGIN RSA PRIVATE KEY-----
MIIEoQIBAACAQBULFg5ujHrtmljnutSuoO8Xe56LlT+HM8v/xkaa39EstM3/
aFXTHgElQijLChp
HungXQ29VtC8rc1bW0lkdi23OH5eqkMHGhvEwqa0HWASUM114o3o/IX+0f2UcPoKCOVUR
+jx71Sg
5AU52EQfanIn3ZQ81FW7Edp5a3q4DhjGlUKToHVbicL5E+g45zfB95wIyywWzfeW/
UUF3LpGZyq/
ebIUlq1qTbHkLbCC2r7RTn8vpQWp47BGVYgtGSBmpTRP5hnbzquj3itkiLHjU39S2sJCJOTrJx5
i8BygR4s3mHKBj81
+ePQxG1kGbf6R4yg6sECmXn17MRQVXODNHZbAgMBAAEcggEAY1tsiUsIwDl5
91CXirkYGuvfLyLflXenxfI50mDFms/mumTqloHO7tr0oriHDR5K7wMcY/
YY5YkcXNo7mvUVDlpM
ZNUJ5r7w9gZRTrf7LylaJ58kOcyajw8TsC4e4LPbFaHwSld6K8rXh64o6WgW4SrsB6ICmrlkGQI7
3wcfgt5ecIu4TZf0OE9IHjn+2eRlrsjBdeORI7KiUNC/pAG23I6MdDOFEQRcCSigCj+4/
mciFUSA
SWS4dMbrpb9FNsIcf9dcLxVM7/6KxgJNfZc9XWzUw77Jg8x92Zd0fVhHOux5IZC
+UvSKWB4dyfcI
tE8C3p9bbU9VgY5vLCAiIb4qQKBgQDLiO24GXrIkswF32YtBBMuVgLGcWu9h9H1O9mKAc2m8Cm1
jUE5IpzRjTedc9I2qiIMUTwtgnw42auSCzbUeYMURPtDqyQ7p6AjMuJp9EPemcSVOK9vXYL0Ptco
xW9MC0dtV6iPkCN7gOqiZXPRKaFbWADp16p8UAIvS/
a5XXk5jwKBgQCKkPHi2EIShluRkxh1jyWC
iDCiK6JBRsMvpLbc0v5dKwP5alo1fmdR5PJaV2qvZSj5CYNpMAy1/EDNTY5OSIJU
+0KFmQbyhsbm
```

```
rdLNL4+TcnT7c62/aH01ohYaf/VCbRhtL1BfqGoQc7+sAc8vmKkesnF7CqCEKdyF/  
dhrxYdQKB  
gCOiZzzNAapayz1+JcVTwwEid6j9JqNXbBc+Z2YwMi+T0Fv/P/hwkX/ypeOXnIUcw0Ih/  
YtGBVAC  
DQbsz7LcY1HqXiHKYNWNvXgwwO  
+oiChjxvEkSdsTTifnK4VScvU9BxDbQHjdINDJbL6oar92UN7V  
rBYvChJZF7LvUH4YmVpHAoGAbZ2X7XvoeEO+uZ58/  
BGKOlGHByHBDiXtzMhdJr15HTYjxK7OgTZm  
gK+8zp4L9IbvLGDMJO8vft32XPEWuvI8twCzFH+CsWLQADZMZKSsBasOZ/h1FwhdMgCMcY  
+Qlzd4  
JZKjTSu3i7vhvx6RzdSedXEMNTZWN4qlIx3kr5aHcukCgYA9T  
+Zrvm1F0seQPbLknn7EqhXIjBaT  
P8TtVw/6bdPi23ExzxZn7KODrfclYRphlLHMPaONv/x2xALIf91UB  
+v5ohy1oDoasL0gijlhouRe  
2ERKKdwz0ZL9SWq6VTdhr/5G994CK72fy5WhyERBDjUIdHaK3M849Jjuf8cSrvSb4g==  
-----END RSA PRIVATE KEY-----
```

4. If you're using OpenSSH (or any reasonably paranoid SSH client), you should set the permissions of this file so it is only readable by you.

On Linux and UNIX, enter the information in the following example.

```
$ chmod 700 id_rsa-gsg-keypair ; ls -l id_rsa-gsg-keypair
```

You receive output similar to the following example.

```
-rw----- 1 fred flintstones 1701 Jun 19 17:57 id_rsa-gsg-keypair
```

How to Add Rules to the Default Security Group

Before you can log in to an instance, you must authorize access.

This section describes how to add rules that allow HTTP access on port 80, SSH access on port 22, and Remote Desktop (RDP) access on port 3389. This enables the instance to be reached on port 80 from the Internet and enables you to administer the instance over SSH or RDP.

AWS Management Console

To authorize access to your instance

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Security Groups** in the **Navigation** pane.
The console displays a list of security groups that belong to the account.
3. Select the **default** security group.
Its rules appear in the lower pane.
4. To add the HTTP rule, enter the following:
 - Select **HTTP** from the **Connection Method** list box.
 - Select **TCP** from the **Protocol** list box.
 - Enter **80** in the **From Port** and **To Port** fields.
 - Enter **0.0.0.0/0** in the **Source** field.

Then, click **Save**.

5. To add the SSH rule, enter the following:
 - Select **SSH** from the **Connection Method** list box.

- Select `TCP` from the **Protocol** list box.
- Enter `22` in the **From Port** and **To Port** fields.
- Enter your public IP address in the **Source** field.

Then, click **Save**.

6. To add the RDP rule, enter the following:
 - Select `RDP` from the **Connection Method** list box.
 - Select `TCP` from the **Protocol** list box.
 - Enter `22` in the **From Port** and **To Port** fields.
 - Enter your public IP address in the **Source** field.

Then, click **Save**.

Command Line Tools

To authorize access to your instance

- Enter the `ec2-authorize` commands.

```
PROMPT> ec2-authorize default -p 22 -s your-public-ip-address/32
PERMISSION    default    ALLOWS    tcp       22        22        FROM     CIDR     your-
public-ip-address/32
PROMPT> ec2-authorize default -p 3389 -s your-public-ip-address/32
PERMISSION    default    ALLOWS    tcp       3389     3389     FROM     CIDR     your-
public-ip-address/32
PROMPT> ec2-authorize default -p 80
PERMISSION    default    ALLOWS    tcp       80        80        FROM     CIDR     0.0.0.0/0
```

Because we didn't specify otherwise, your instance was launched in your `default` group. The first command authorizes network access to instances in your default group on the standard SSH port (22). Similarly, the second command opens up the standard HTTP port (80).

How to Run an Instance

This section describes how to run an instance.

AWS Management Console

To launch an instance

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Instances** in the **Navigation** pane.
The console displays a list of running instances.
3. Click **Launch Instance**.
The Launch Instance wizard appears.
4. Select the **Quick Start** tab.
5. If you are launching a Linux/UNIX instance, locate the **Getting Started on Fedora Core 8** AMI and click its **Select** button. If you are launching a Windows instance, locate the **Getting Started on Microsoft Windows Server 2003** AMI and click its **Select** button.



Note

We recommend launching basic AMIs for this tutorial, but you can launch any AMI.

- If the **Configure Firewall** page of the wizard appears, click the **Skip** button because you already configured the `default` security group.
The **Launch** page of the wizard appears.
- Confirm the following settings and click **Launch**.
 - Enter `1` in the **Number of Instances** field.
 - Select the `m1.small` **Instance Type** option.
 - Select the key pair that you created from the **Key Pair Name** list box.
 - Select `default` from the **Security Groups** list box.

The instance(s) begin launching.

Command Line Tools

To launch an instance

- Use the `ec2-run-instances` command.

```
PROMPT> ec2-run-instances ami-235fba4a -k gsg-keypair
```

Amazon EC2 returns output similar to the following example.

```
RESERVATION    r-7430c31d      924417782495    default
INSTANCE       i-ae0bf0c7      ami-2bb65342    pending gsg-keypair  0
  m1.small     2008-03-21T16:19:25+0000    us-east-1a
```

- Look for the instance ID in the second field and write it down.
You use it to manipulate this instance (including terminating it when you are finished).
It takes a few minutes for the instance to launch.
- The following command displays the launch status of the instance.

```
PROMPT> ec2-describe-instances i-ae0bf0c7
RESERVATION    r-7430c31d      924417782495    default
INSTANCE       i-ae0bf0c7      ami-2bb65342
ec2-67-202-7-236.compute-1.amazonaws.com
ip-10-251-31-162.ec2.internal    running gsg-keypair  0
  m1.small     2008-03-21T16:19:25+0000us-east-1a
```



Important

After launching an instance, you are billed hourly for running time. When you are finished, make sure to terminate any instances that you started.

When the instance state in the field just before the key pair name reads "running" the instance started booting. There might be a short time before it is accessible over the network, however. The first DNS name is your instance's external DNS name, i.e. the one that can be used to contact it from the Internet. The second DNS name is your instance's local DNS name, and is only contactable by other instances within the Amazon EC2 network. The DNS names of your instances are different than those shown in the preceding example and you should use yours instead. The examples in this guide use the public DNS name.

Instance Metadata

Amazon EC2 instances can access instance-specific metadata as well as data supplied when launching the instances. This data can be used to build more generic AMIs that can be modified by configuration files supplied at launch time.

If you run web servers for various small businesses, they can all use the same AMI and retrieve their content from the Amazon S3 bucket you specify at launch.

To add a new customer at any time, simply create a bucket for the customer, add their content, and launch your AMI.



Note

For information on categories of metadata, see [Metadata Categories \(p. 137\)](#).

Data Retrieval

An instance retrieves the data by querying a web server using a Query API. The base URI of all requests is `http://169.254.169.254/2009-04-04/` where `2009-04-04` indicates the API version.



Note

Amazon EC2 Version 1.0 is part of a legacy versioning scheme. Newer versions follow a date based versioning scheme. For more information on the versioning scheme used by Amazon EC2, go to the [Amazon Elastic Compute Cloud API Reference](#).

The latest version of the API is always available using the URI `http://169.254.169.254/latest`.

Security of Launch Data

Although only your specific instance can access launch data, the data is not protected by cryptographic methods. You should take suitable precautions to protect sensitive data (such as long lived encryption keys).



Note

You are not billed for HTTP requests used to retrieve metadata and user-supplied data.

Metadata Retrieval

Requests for a specific metadata resource returns the appropriate value or a 404 HTTP error code if the resource is not available. All metadata is returned as text (content type `text/plain`).

Requests for a general metadata resource (i.e. an URI ending with a `/`) return a list of available resources or a 404 HTTP error code if there is no such resource. The list items are on separate lines terminated by line feeds (ASCII 10).

Example

The following examples list HTTP GET requests and responses. You can use a tool such as curl or wget to make these types of requests.

This example gets the available API versions.

```
GET http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2009-04-04
```

This example gets the top-level metadata items.

```
GET http://169.254.169.254/2009-04-04/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
instance-id
instance-type
local-hostname
local-ipv4
placement/
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
```

This example gets the value of each metadata item from the preceding example.

```
GET http://169.254.169.254/2009-04-04/meta-data/ami-manifest-path
my-amis/spamd-image.manifest.xml
GET http://169.254.169.254/2009-04-04/meta-data/ami-manifest-path
my-amis/spamd-image.manifest.xml
GET http://169.254.169.254/2009-04-04/meta-data/ami-id
ami-2bb65342
GET http://169.254.169.254/2009-04-04/meta-data/reservation-id
r-fea54097
GET http://169.254.169.254/2009-04-04/meta-data/hostname
ec2-67-202-51-223.compute-1.amazonaws.com
```

This example gets the list of available public keys.

```
GET http://169.254.169.254/2009-04-04/meta-data/public-keys/
0=my-public-key
```

This example shows the formats in which public key 0 is available.

```
GET http://169.254.169.254/2009-04-04/meta-data/public-keys/0/
openssh-key
```

This example gets public key 0 (in the OpenSSH key format).

```
GET http://169.254.169.254/2009-04-04/meta-data/public-keys/0/openssh-key
ssh-rsa AAAA.....wZef my-public-key
```

This example gets the product code.

```
GET http://169.254.169.254/2009-04-04/meta-data/product-codes
774F4FF8
```

User Data Retrieval

Requests for the user data returns the data as-is (content type `application/x-octetstream`).



Note

All user-supplied data is treated as opaque data; what you give us is what you get back. It is the responsibility of the instance to interpret this data appropriately.

Example

This shows an example of returning comma-separated user-supplied data.

```
GET http://169.254.169.254/2009-04-04/user-data
1234,fred,reboot,true | 4512,jimbo, | 173,, ,
```

This shows an example of returning line-separated user-supplied data.

```
GET http://169.254.169.254/2009-04-04/user-data
[general]
instances: 4

[instance-0]
s3-bucket: fred

[instance-1]
reboot-on-error: yes
```

Use Case: AMI Launch Index Value

In this example, Alice wants to launch four instances of her favorite database AMI with the first acting as master and the remainder acting as replicas.

The master database configuration specifies various database parameters (e.g., the size of store) while the replicas' configuration specifies different parameters, such as the replication strategy. Alice decides to provide this data as an ASCII string with a pipe symbol (|) delimiting the data for the various instances:

```
store-size=123PB backup-every=5min | replicate-every=1min | replicate-
every=2min | replicate-every=10min | replicate-every=20min
```

The `store-size=123PB backup-every=5min` defines the master database configuration, `replicate-every=1min` defines the first replicant's configuration, `replicate-every=2min` defines the second replicant's configuration, and so on.

Alice launches four instances.

```
PROMPT> ec2-run-instances ami-2bb65342 -n 4 -d "store-size=123PB backup-
every=5min | replicate-every=1min | replicate-every=2min | replicate-
every=10min | replicate-every=20min"
```

```
RESERVATION    r-fea54097          598916040194    default
INSTANCE i-3ea74257  ami-2bb65342  pending 0 m1.small 2007-08-07T11:29:58+0000
us-east-1c
INSTANCE i-31a74258  ami-2bb65342  pending 1 m1.small 2007-08-07T11:29:58+0000
us-east-1c
INSTANCE i-31a74259  ami-2bb65342  pending 2 m1.small 2007-08-07T11:29:58+0000
us-east-1c
INSTANCE i-31a7425a  ami-2bb65342  pending 3 m1.small 2007-08-07T11:29:58+0000
us-east-1c
```

Once launched, all instances have a copy of the user data and the common metadata shown here:

- AMI id: ami-2bb65342
- AMI manifest path: ec2-public-images/getting-started.manifest.xml
- Reservation ID: r-fea54097
- Public keys: none
- Security group names: default
- Instance type: m1.small

However each instance has certain unique metadata.

Instance 1

Metadata	Value
instance-id	i-3ea74257
ami-launch-index	0
public-hostname	ec2-67-202-51-223.compute-1.amazonaws.com
public-ipv4	67.202.51.223
local-hostname	ip-10-251-50-35.ec2.internal
local-ipv4	10.251.50.35

Instance 2

Metadata	Value
instance-id	i-31a74258
ami-launch-index	1
public-hostname	ec2-67-202-51-224.compute-1.amazonaws.com
public-ipv4	67.202.51.224
local-hostname	ip-10-251-50-36.ec2.internal
local-ipv4	10.251.50.36

Instance 3

Metadata	Value
instance-id	i-31a74259
ami-launch-index	2
public-hostname	ec2-67-202-51-225.compute-1.amazonaws.com
public-ipv4	67.202.51.225
local-hostname	ip-10-251-50-37.ec2.internal
local-ipv4	10.251.50.37

Instance 4

Metadata	Value
instance-id	i-31a7425a
ami-launch-index	3
public-hostname	ec2-67-202-51-226.compute-1.amazonaws.com
public-ipv4	67.202.51.226
local-hostname	ip-10-251-50-38.ec2.internal
local-ipv4	10.251.50.38

Therefore, an instance can determine its portion of the user-supplied data through the following process.

Metadata Discovery Process

1	Determine the instance in the launch group. <code>GET http://169.254.169.254/2009-04-04/meta-data/ami-launch-index 1</code>
2	Retrieve the user data. <code>GET http://169.254.169.254/2009-04-04/user-data store-size=123PB backup-every=5min replicate-every=1min replicate-every=2min replicate-every=10min replicate- every=20min</code>
3	Extract the appropriate part of the user data. <code>user_data.split(' ')[ami_launch_index]</code>

Instance Storage

Every instance includes a fixed amount of storage space on which you can store data. Within this document, it is referred to as the "instance store" as it is not designed to be a permanent storage solution.

If an instance reboots (intentionally or unintentionally), the data on the instance store will survive. If the underlying drive fails or the instance is terminated, the data will be lost.

We highly recommend backing up important data to Amazon S3.

Making Instance Stores Available

Inside the instance, instance stores are exposed as normal block devices and can be formatted as any file system and mounted.

Making Instance Stores Available in Linux and UNIX

Depending on the instance type, some instance stores are not mounted or formatted. To mount and format an instance store, use the Linux and UNIX `mount` and `mkfs` commands.

For added security and safety, we recommend using an encrypted file system.

Making Instance Stores Available in Windows

To initialize a volume

1. Log in to your instance using Remote Desktop.
2. On the taskbar, click **Start**, and then click **Run**.
3. Type `diskmgmt.msc` and click **OK**. The Disk Management utility opens.
4. Right-click the Amazon EBS volume, select **Initialize**, and follow the on-screen prompts.



Note

If the **Initialize** option does not appear, select **Format**.

Disk Performance Optimization

Due to how Amazon EC2 virtualizes disks, the first write to any location on an instance's drives performs slower than subsequent writes. For most applications, amortizing this cost over the lifetime of the instance is acceptable. However, if you require high disk performance, we recommend initializing drives by writing once to every drive location before production use.

To initialize the stores, use the following commands on the m1.large, m1.xlarge, and c1.xlarge instance types:

```
dd if=/dev/zero of=/dev/sdb bs=1M
dd if=/dev/zero of=/dev/sdc bs=1M
dd if=/dev/zero of=/dev/sdd bs=1M (m1.xlarge only)
dd if=/dev/zero of=/dev/sde bs=1M (m1.xlarge only)
```

To perform the initialization on all drives at the same time, use the following command:

```
dd if=/dev/zero bs=1M|tee /dev/sdb|tee /dev/sdc|tee /dev/sde > /dev/sdd
```



Note

Make sure to unmount the drive before performing this command..

Initialization can take a long time (about 8 hours for an extra large instance).

RAID Configuration

Configuring drives for RAID initializes them by writing to every drive location. When configuring software-based RAID, make sure to change the minimum reconstruction speed:

```
echo $((30*1024)) > /proc/sys/dev/raid/speed_limit_min
```



Note

You cannot use `iostat` (part of the `sar` System Activity Reporting package) to watch performance. You also cannot watch `'cat /proc/mdstat'`.

Using Shared AMIs

This section describes how to find and safely use shared AMIs. One of the easiest ways to get started with Amazon EC2 is to use a shared AMI that has the components you need and add custom content.

How to Find Shared AMIs

To find shared AMIs

- Enter the `ec2-describe-images` command (or the abbreviated `ec2dim` command) with a flag to filter the results.

Example

This command displays a list of all public AMIs.

```
PROMPT> ec2dim -x all
```

The `-x all` flag shows AMIs executable by all users. This includes AMIs you own.

This command displays a list of AMIs for which you have explicit *launch permissions*.

```
PROMPT> ec2dim -x self
```

AMIs that you own are excluded from the list.

This command displays a list of AMIs owned by Amazon.

```
PROMPT> ec2dim -o amazon
```

This command displays a list of AMIs owned by a particular user.

```
PROMPT> ec2dim -o <target_uid>
```

The `<target_uid>` is the account ID of the user who owns the AMIs for which you are looking.

Safe Use of Shared AMIs

AMIs are launched at the user's own risk. Amazon cannot vouch for the integrity or security of AMIs shared by other 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 (a web site, another user, etc). If you do not know the source of an AMI, we recommended that you search the 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.

If you plan to use a shared AMI, review the following table to confirm the AMI is not doing anything malicious.

Launch Confirmation Process

1	Check the <code>ssh authorized keys</code> file. The only key in the file should be the key you used to launch the AMI.
---	---

2	Check open ports and running services.
3	Change the root password if is not randomized on startup. For more information on randomizing the root password on startup, see Disable Password-Based Logins for Root (p. 50) .
4	Check if ssh allows root password logins. See Disable Password-Based Logins for Root (p. 50) for more information on disabling root based password logins.
5	Check whether there are any other user accounts that might allow backdoor entry to your instance. Accounts with super user privileges are particularly dangerous.
6	Verify that all cron jobs are legitimate.

Paying for AMIs

- [How to Find Paid AMIs \(p. 78\)](#)
- [Purchasing a Paid AMI \(p. 79\)](#)
- [How to Launch Paid AMIs \(p. 79\)](#)
- [Paid Support \(p. 80\)](#)
- [Bills for Paid and Supported AMIs \(p. 81\)](#)

Amazon EC2 integrates with Amazon DevPay, allowing developers to charge 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](#). For more information about charging for your use of your AMIs, or providing support, see [Creating Paid AMIs \(p. 56\)](#)

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.

How to 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 Amazon EC2 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:

```
PROMPT> ec2-describe-images <ami_id>
```

The `<ami_id>` is the AMI ID.

The command returns the following:

```
IMAGE <ami_id> <manifest> <user_id>, <status> {private | public}  
<product_code>
```

The `<ami_id>` is the AMI ID, `<manifest>` is the manifest location, `<user_id>` is the ID of the user that owns the AMI, `<status>` indicates whether the AMI is available, and `<product_code>` is the product code associated with the AMI. If a product code is present, the AMI is a paid AMI.

Example

This example shows an **ec2-describe-images** call describing a paid AMI. The product code is 774F4FF8.

```
PROMPT> ec2-describe-images ami-2bb65342  
IMAGE ami-2bb65342 awesome-ami/webserver.manifest.xml AIDADH4IGTRXXKCD  
available private 774F4FF8
```

Purchasing 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, see [Reserved Instance Concepts \(p. 14\)](#).

How to Launch 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 (which will be more than the base Amazon EC2 rate).

To launch a paid AMI

- Enter the following command:

```
PROMPT> ec2-run-instances <ami_id>
```

The `<ami_id>` is the AMI ID.



Note

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

Example

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

```
PROMPT> ec2-run-instances ami-2bb65342
RESERVATION r-a034c7c9 924417782495 default
INSTANCE i-400df629 ami-2bb65342 pending 0 m1.small 2008-03-21T18:49:33+0000
us-east-1c
```

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, see [Reserved Instance Concepts \(p. 14\)](#).

To associate the product code with your AMI

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

```
PROMPT> 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 the `run-instances`. The instance is charged according to the rates set by the AMI owner.

Example

The following command associates the `ami-2bb65342` AMI with the `774F4FF8` product code.

```
PROMPT> ec2-modify-image-attribute ami-2bb65342 --product-code 774F4FF8
productCodes      ami-2bb65342      productCode      774F4FF8
```

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

```
PROMPT> ec2-run-instances ami-2bb65342
RESERVATION r-a034c7c9 924417782495 default
INSTANCE i-400df629 ami-2bb65342 pending 0 m1.small 2008-03-21T18:49:33+0000
us-east-1c
```

Bills for Paid and Supported AMIs

At the end of each month, you receive an e-mail 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.

At any time, you can view the usage information for your paid and supported AMIs (go to <http://www.amazon.com/dp-applications>).

Getting Console Output and Rebooting Instances

Console output is a valuable tool for problem diagnosis. It is especially useful for troubleshooting kernel problems and service configuration issues that could cause an instance to terminate or become unreachable before its SSH daemon can be started.

Similarly, the ability to reboot instances that are otherwise unreachable is valuable for both troubleshooting and general instance management.

Amazon EC2 instances do not have a physical monitor through which you can view their console output. They also lack physical controls that allow you to power up, reboot, or shut them down. To allow these actions, we provide them through the Amazon EC2 SOAP API, Query API, and command line tools.

Console Output

For Linux and UNIX instances, the Amazon EC2 instance console output displays the exact console output that would normally be displayed on a physical monitor attached to a machine. This output is buffered because the instance produces it and then posts it to a store where the instances owner can retrieve it.

For Windows instances, the Amazon EC2 instance console output displays the last three system event log errors.

The posted output is not continuously updated; only when it is likely to be of the most value. This includes shortly after instance boot, after reboot, and when the instance terminates.



Note

Only the most recent 64 KB of posted output is stored, which is available for at least 1 hour after the last posting.

You can retrieve the console output for an instance using `GetConsoleOutput`. For more information, go to the [Amazon Elastic Compute Cloud API Reference](#) or [Amazon Elastic Compute Cloud Command Line Reference](#).



Note

Only the instance owner can access the console output.

Instance Reboot

Just as you can reset a machine by pressing the reset button, you can reset Amazon EC2 instances using `RebootInstances`. For more information, go to the [Amazon Elastic Compute Cloud API Reference](#) or [Amazon Elastic Compute Cloud Command Line Reference](#).



Caution

For Windows instances, this operation performs a hard reboot that might result in data corruption.

Related Topics

- [Amazon EC2 Flow \(p. 16\)](#)
- [AMI and Instance Concepts \(p. 10\)](#)

Accessing Instances

Topics

- [Accessing Instances in Linux and UNIX \(p. 83\)](#)
- [Accessing Instances in Windows \(p. 84\)](#)
- [Related Topics \(p. 86\)](#)

This section describes how to access instances that you launched. For information on launching instances, see [Launching and Using Instances \(p. 64\)](#)

Accessing Instances in Linux and UNIX

This section describes how to access Linux and UNIX instances using SSH.

How to Authorize Network Access to Your Instances

Before accessing your instance, you must authorize access.

To authorize access to your instance

1. Enter the `ec2-authorize` command to allow all IP addresses to access your instance through the port 80 (public web) IP address.

```
PROMPT> ec2-authorize default -p 80
PERMISSION    default  ALLOWS  tcp    80      80      FROM    CIDR
0.0.0.0/0
```

2. Get the public IP address of your local machine by going to a search engine, entering "what is my IP address," and using one of the provided services.
3. Enter the `ec2-authorize` command to open port 22 (SSH port) to your IP address.

```
PROMPT> ec2-authorize default -p 22 -s your_ip_address/32
PERMISSION    default  ALLOWS  tcp    22      22      FROM    CIDR
your_ip_address/32
```

This command allows access from your IP address only. If your IP address is dynamic, you need to use this command each time it changes. To allow additional IP address ranges, use this command for each range.

How to Connect to your Instance

This section describes how to connect to your instance.

To connect to your instance

1. Open a web browser and go to `http://<hostname>/`, where `<hostname>` is your instance's public hostname as returned by `ec2-describe-instances` (`ec2-67-202-51-223.compute-1.amazonaws.com` in the example). A webpage welcoming you to your instance displays.



Note

If the web site times out, your instance might not have finished starting up. Wait a couple of minutes and try again.

2. Whenever you launch a public AMI that you have not rebundled, run the `ec2-get-console-output` command and locate the SSH HOST KEY FINGERPRINTS section.

```
PROMPT> ec2-get-console-output instance_id

...
ec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----
ec2: 2048 bc:89:29:c6:45:4b:b3:e2:c1:41:81:22:cb:3c:77:54
   /etc/ssh/ssh_host_key.pub
ec2: 2048 fc:8d:0c:eb:0e:a6:4a:6a:61:50:00:c4:d2:51:78:66
   /etc/ssh/ssh_host_rsa_key.pub
ec2: 1024 b5:cd:88:6a:18:7f:83:9d:1f:3b:80:03:10:17:7b:f5
   /etc/ssh/ssh_host_dsa_key.pub
ec2: -----END SSH HOST KEY FINGERPRINTS-----
...
```

Note the fingerprints. You will need to compare them in the next step.

3. Use the following command to login as root and exercise full control over this instance as you would any host.

```
$ ssh -i id_rsa-gsg-keypair
root@ec2-67-202-51-223.compute-1.amazonaws.com
The authenticity of host 'ec2-67-202-51-223.compute-1.amazonaws.com
(216.182.225.42)' can't be established.
RSA key fingerprint is fc:8d:0c:eb:0e:a6:4a:6a:61:50:00:c4:d2:51:78:66.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added
'ec2-67-202-51-223.compute-1.amazonaws.com' (RSA) to the list of known
hosts.
Last login: Wed Jun 21 08:02:08 2006
root@ec2-67-202-51-223 #
```

If you are launching a public AMI, verify the fingerprint matches one of the fingerprints from the output of the `ec2-get-console-output` command. If it doesn't, someone might be attempting a "man-in-the-middle" attack.



Note

Your machine might have a different name for the preceding `ssh` command or use different command line options.

Accessing Instances in Windows

This section describes how to access instances running Windows.

How to Get the Instance Password

To access an instance, you must get its administrator password.



Note

The Windows password is only generated the first time an AMI is launched. It is not generated for rebundled AMIs or after the password is changed on an instance.

The password is encrypted using the key pair that you provided and stored within the `<password>` tags of the console output.

To get the administrator password

1. Enter the following command:

```
PROMPT> ec2-get-password -k gsg-keypair
```

Amazon EC2 returns the Windows password.



Note

Unless the key pair is in the current directory, you must specify the full key pair path.

2. Write down or save the password.
After logging in, you can change it.

How to Get the RDP Certificate

To verify you are connecting to the correct server, you must get the RDP certificate.

To get the RDP certificate

1. Enter the following command:

```
PROMPT> ec2-get-console-output instance_id
```

```
date_time <RDPCERTIFICATE><DN>CN=instance_id, OU=EC2, O=Amazon.com</DN><THUMBPRINT>rdp_certificate</THUMBPRINT></RDPCERTIFICATE>
```

Amazon EC2 returns the RDP certificate information.

2. Write down or save the information.

How to Access the Instance

This section describes how to connect to your instance.

To connect to your instance

1. Get the public IP address of your local machine by going to a search engine, entering "what is my IP address," and using one of the provided services.
2. Authorize the security group to allow Remote Desktop access:

```
PROMPT> ec2-authorize default -p 3389 -s your_ip_address/32  
PERMISSION    default  ALLOWS  tcp      3389     3389     FROM     CIDR  
              your_ip_address/32
```

3. Retrieve the FQDN of your instance.

This example retrieves the FQDN of the `i-ae0bf0c7` instance.

```
PROMPT> ec2-describe-instances i-ae0bf0c7  
RESERVATION  r-7430c31d  924417782495  default  
INSTANCE     i-ae0bf0c7  ami-2bb65342  
ec2-67-202-7-236.compute-1.amazonaws.com  ip-10-251-31-162.ec2.internal  
running     gsg-keypair  0             m1.small  
2008-03-21T16:19:25+0000  us-east-1a
```

In this example, the FQDN is `ec2-67-202-7-236.compute-1.amazonaws.com`

4. On the taskbar, click **Start**, point to **Programs**, point to **Accessories**, point to **Communications**, and click **Remote Desktop Connection**.

The **Remote Desktop Connection** dialog box appears.

- a. Enter the FQDN in the **Computer** field.
- b. Click the **Advanced** or **Security** tab.
- c. Select **Warn me** or **Attempt authentication** from the list box.
- d. Click **Connect**.

The Amazon EC2 instance returns a security alert.

5. To verify the instance, click **View Certificate**.

The **Certificate** page appears.

6. Click the **Details** tab.

The **Details** page appears.

7. Select the **Thumbprint** and verify it against the value you wrote down in previous procedure.

8. If it matches, click **OK** and then **Yes**.

The Remote Desktop Connection client connects to the instance.

9. Enter "administrator" as the user name and the instance password. .

You can now use the Amazon EC2 instance as you would any Windows-based system.



Note

If you plan to allow other users to remotely access the instance, you must add them to the Remote Desktop Users group.

Related Topics

- [Amazon EC2 Flow \(p. 16\)](#)
- [AMI and Instance Concepts \(p. 10\)](#)

Using Instance Addressing

Topics

- [API Overview](#) (p. 87)
- [Determining Your IP Addresses](#) (p. 87)
- [Using Elastic IP Addresses](#) (p. 88)

This section describes how to perform common instance addressing tasks.

API Overview

This section provides a brief overview of each elastic IP address operation.

- **AllocateAddress**—Acquires an elastic IP address for use with your account.
- **DescribeAddresses**—Lists elastic IP addresses assigned to your account.
- **ReleaseAddress**—Releases an elastic IP address associated with your account.
After releasing an elastic IP address, it is released to the IP address pool and might no longer be available to your account.
- **AssociateAddress**—Associates an elastic IP address with an instance.
- **DisassociateAddress**—Disassociates the specified elastic IP address from the instance to which it is assigned.

Determining Your IP Addresses

This section describes how to determine your internal and external IP addresses.

AWS Management Console

To determine your private and public IP addresses

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Instances** in the **Navigation** pane.
The console displays a list of running instances.
3. Locate and select an instance.
The console displays information about the instance in the lower pane.
4. To determine the public IP address, use the IP address specified within the **Public DNS** field.
5. To determine the private IP address, use the IP address specified within the **Private DNS** field.

Command Line Tools

To determine your private IP address in Linux and UNIX

1. Connect to the instance.
2. Enter one of the following commands:
 - `# ifconfig eth0`
 - `# curl http://169.254.169.254/latest/meta-data/local-ipv4`

The second option refers to the instance data. For more information, see [Instance Metadata](#) (p. 71).

To determine your private IP address in Windows

1. Connect to the instance.
2. On the taskbar, click **Start**, right-click **My Computer**, and select **Properties**.
3. Click the **Computer Name** tab. The IP address appears in the **Full computer name** field.

To determine your public IP address

1. Connect to the instance.
2. Determine your public IP address from your instance by referring to the instance data.

```
PROMPT> curl http://169.254.169.254/latest/meta-data/public-ipv4
```

Using Elastic IP Addresses

Elastic IP addresses are static IP addresses designed for dynamic cloud computing. An elastic IP address is associated with your account, not a particular instance. You control addresses associated with your account until you choose to explicitly release them.

This section describes how to perform common elastic IP address tasks.

Allocating Elastic IP Addresses

This section describes how to assign an Amazon EC2 elastic IP address to your account and verify it.

AWS Management Console

To allocate a new IP address for use with your account

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Elastic IPs** in the **Navigation** pane.
The console displays a list of elastic IP addresses assigned to your account.
3. Click **Allocate New Address**.
A confirmation dialog box appears.
4. Click **Yes, Allocate**.
A new elastic IP address appears in the list.

Command Line Tools

To allocate a new IP address for use with your account

- Enter the following command:

```
PROMPT> ec2-allocate-address
```

Amazon EC2 returns an elastic IP address similar to the following:

```
ADDRESS 75.101.155.119
```



Note

An Elastic IP address is associated with an account and billed accordingly until the address is released using `ec2-release-address` command.

Describing Elastic IP Addresses

This section describes how to view the elastic IP addresses allocated to your account.

AWS Management Console

To view elastic IP addresses assigned to your account

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Elastic IPs** in the **Navigation** pane.
The console displays a list of elastic IP addresses assigned to your account.
3. To reduce the size of the list, start typing part of the IP address or instance ID to which it is assigned in the search box.

Command Line Tools

To view elastic IP addresses assigned to your account

1. To view all elastic IP addresses assigned to your account::

```
PROMPT> ec2-describe-addresses
```

Amazon EC2 returns a list of elastic IP addresses similar to the following:

```
ADDRESS 75.101.157.145  
ADDRESS 75.101.155.119
```

2. To verify a specific elastic IP address:

```
PROMPT> ec2-describe-addresses ip_address
```

Amazon EC2 returns the specified elastic IP address, similar to the following:

```
ADDRESS 75.101.157.145
```

Associating an Elastic IP Address with a Running Instance

Once an elastic IP address is allocated, you can map it to a running instance.

AWS Management Console

To associate an elastic IP address with an instance

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Instances** in the **Navigation** pane.
The console displays a list of running instances.
3. Write down the instance ID to associate with the elastic IP address.
4. Click **Elastic IPs** in the **Navigation** pane.
The console displays a list of elastic IP addresses assigned to your account.
5. Select an instance and click **Associate**.

The **Associate Address** dialog box appears.

6. Select the instance from the **Instance ID** list box and click **Associate**.
The elastic IP address is associated with the instance.

Command Line Tools

To associate an elastic IP address with an instance

1. Describe running instances:

```
PROMPT> ec2-describe-instances
```

Amazon EC2 returns output similar to the following:

```
RESERVATION      r-ae33c2c7      924417782495      default
INSTANCE         i-b2e019da      ami-2bb65342
ec2-72-44-33-67.compute-1.amazonaws.com ip-10-251-71-165.ec2.internal
  running gsg-keypair 0          ml.small
2008-03-03T23:09:09+0000      us-east-1a
INSTANCE         i-b2e019db      ami-2bb65342
ec2-67-202-3-83.compute-1.amazonaws.com ip-10-251-47-36.ec2.internal
  running gsg-keypair 1          ml.small
2008-03-03T23:09:09+0000      us-east-1a
```

Write down the instance ID to associate with an elastic IP address.

2. Describe elastic IP addresses assigned to the account:

```
PROMPT> ec2-describe-addresses
```

Amazon EC2 returns a list of elastic IP addresses similar to the following:

```
ADDRESS 75.101.157.145
ADDRESS 75.101.155.119
```

Write down the elastic IP address to associate with an instance.

3. To associate the instance and elastic IP address:

```
PROMPT> ec2-associate-address -i instance_id ip_address
```

Amazon EC2 returns output similar to the following:

```
ADDRESS 75.101.157.145 i-b2e019da
```

4. Associations take a few minutes to complete. To verify the association using `ec2-describe-addresses`:

```
PROMPT> ec2-describe-addresses
```

Amazon EC2 returns output similar to the following:

```
ADDRESS 75.101.157.145 i-b2e019da
```

5. To verify the association using `ec2-describe-instances`:

```
PROMPT> ec2-describe-instances-i instance_id
```

Amazon EC2 returns output similar to the following:

```
RESERVATION      r-ae33c2c7      924417782495      default
INSTANCE         i-b2e019da      ami-2bb65342
ec2-75-101-157-145.compute-1.amazonaws.com ip-10-251-71-165.ec2.internal
```

```
running gsg-keypair      0      m1.small
2008-03-03T23:09:09+0000  us-east-1a
```

Associating an Elastic IP Address with a Different Running Instance

Once an Elastic IP Address is allocated, you can map it to a different running instance.



Note

It is highly unlikely that an instance will be configured with its original public IP address that it used prior to being mapped.

AWS Management Console

To remap an IP address

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Instances** in the **Navigation** pane.
The console displays a list of running instances.
3. Write down the new instance ID to associate with the elastic IP address.
4. Click **Elastic IPs** in the **Navigation** pane.
The console displays a list of elastic IP addresses assigned to your account.
5. Locate the IP address to remap and click **Disassociate**.
A confirmation dialog box appears.
6. Click **Yes, Disassociate**.
The elastic IP address is disassociated from the instance and you are returned to the list of elastic IP addresses assigned to your account.
7. Locate the IP address in the list and click **Associate**.
The **Associate Address** dialog box appears.
8. Select the new instance from the **Instance ID** list box and click **Associate**.
The elastic IP address is associated with the new instance.

Command Line Tools

To remap an IP address

1. Describe running instances:

```
PROMPT> ec2-describe-instances
```

Amazon EC2 returns output similar to the following:

```
RESERVATION      r-ae33c2c7      924417782495      default
INSTANCE         i-b2e019da      ami-2bb65342
ec2-67-202-46-87.compute-1.amazonaws.com  ip-10-251-71-165.ec2.internal
  running gsg-keypair      0      m1.small
2008-03-03T23:09:09+0000  us-east-1a
INSTANCE         i-b2e019db      ami-2bb65342
ec2-75-101-157-145.compute-1.amazonaws.com ip-10-251-47-36.ec2.internal
  running gsg-keypair      1      m1.small
2008-03-03T23:09:09+0000  us-east-1a
```

Write down the instance ID to associate with an elastic IP address.

- Associate the address with a new instance:

```
PROMPT> ec2-associate-address -i instance_id ip_address
```

Amazon EC2 returns output similar to the following:

```
ADDRESS 75.101.157.145 i-b2e019da
```

- Verify the changes:

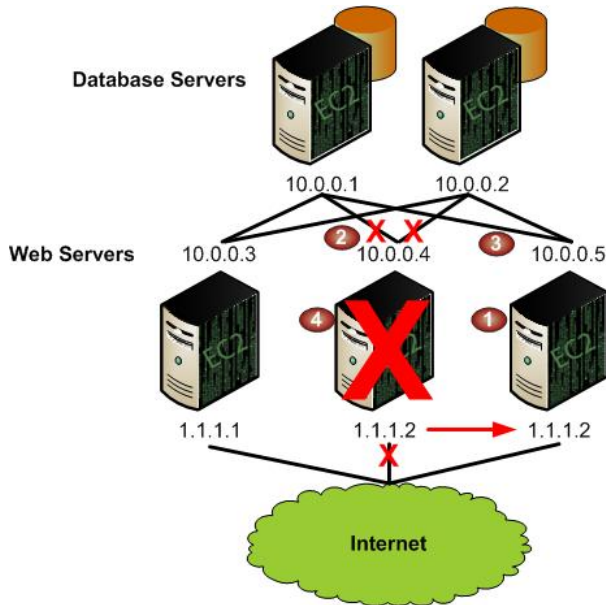
```
PROMPT> ec2-describe-instances
```

Amazon EC2 returns output similar to the following:

```
RESERVATION    r-ae33c2c7    924417782495    default
INSTANCE       i-b2e019da    ami-2bb65342
                ec2-75-101-157-145.compute-1.amazonaws.com
                ip-10-251-71-165.ec2.internal    running    gsg-keypair    0
m1.small       2008-03-03T23:09:09+0000    us-east-1a
INSTANCE       i-b2e019db    ami-2bb65342
                ec2-67-202-46-39.compute-1.amazonaws.com    ip-10-251-47-36.ec2.internal
                running    gsg-keypair    1            m1.small
                2008-03-03T23:09:09+0000    us-east-1a
```

Example

In the following example, web servers are connected to the Internet through elastic IP addresses and to database servers through their private IP addresses.



The administrator decides to replace a web server with a larger instance type. To do this, the administrator starts a new instance using a larger instance type (1), disassociates an elastic IP address from a running instance (2), associates the elastic IP address with the new instance (3), and terminates the old instance (4).

AWS Management Console Example

The following demonstrates how to set up these tasks using the AWS Management Console.

Launch Process

1	The user clicks Instances in the Navigation pane. The console displays a list of running instances.
2	The user clicks Launch Instance , specifies settings that meet the new requirements, and clicks Launch .
3	The user clicks Elastic IPs in the Navigation pane. The console displays a list of elastic IP addresses assigned to the account.
4	The user selects the desired IP address that is assigned to another instance, clicks Disassociate , and confirms the disassociation. The user is returned to the list of elastic IP addresses.
5	The user locates the IP address in the list, clicks Associate , selects the new instance, and confirms the association. The elastic IP address is assigned to the new instance.
6	The user clicks Instances in the Navigation , selects the old instance, and clicks Terminate . Amazon EC2 begins shutting down the old instance.

Command Line Tools Example

The following demonstrates how to set up these tasks using the command line tools.

```
PROMPT> ec2-run-instances ami-6ba54002 -n 1 --availability-zone us-east-1a
RESERVATION r-a034c7c9 924417782495 default
INSTANCE i-3ea74257 ami-6ba54002 pending 0 m1.large 2007-07-11T16:40:44+0000
us-east-1a
```

```
PROMPT> ec2-disassociate-address 67.202.55.255
ADDRESS 67.202.55.255
```

```
PROMPT> ec2-associate-address -i i-3ea74257 67.202.55.255
ADDRESS 67.202.55.255 i-43a4412a
```

```
PROMPT> ec2-terminate-instances i-4bc32334
INSTANCE i-4bc32334 running shutting-down
```

Related Topics

- [Instance Addressing Concepts \(p. 17\)](#)

Using Network Security

Topics

- [API Overview](#) (p. 94)
- [Creating a Security Group](#) (p. 94)
- [Describing Security Groups](#) (p. 95)
- [Adding a Security Group Rule](#) (p. 95)
- [Delete a Security Group Rule](#) (p. 96)
- [Delete a Security Group](#) (p. 97)
- [Example](#) (p. 97)

This section describes how to use Amazon EC2 network security.



Note

In addition to these examples, you can maintain your own firewall on any of your instances. This can be useful if you have specific requirements not met by the Amazon EC2 distributed firewall.

API Overview

This section provides a brief overview of each operation.

- **CreateSecurityGroup**—Creates a new security group for use with your account.
- **DescribeSecurityGroups**—Returns information about security groups associated with your account.
- **DeleteSecurityGroup**—Deletes security groups associated with your account.
- **AuthorizeSecurityGroupIngress**—Adds permissions to a security group.
- **RevokeSecurityGroupIngress**—Revokes permissions from a security group.

Creating a Security Group

This section describes how to create a security group.

AWS Management Console

To create a security group

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Security Groups** in the **Navigation** pane.
The console displays a list of current security groups.
3. Click **Security Group**.
The **Security Group** dialog box appears.
4. Configure the following settings and click **Create**.
 - Security Group Name
 - Security Group Description

Amazon EC2 begins creating the security group.

Command Line Tools

To create a security group

- Enter the following command:

```
PROMPT> ec2-add-group groupname -d "group_description"
```

Amazon EBS returns information similar to the following example.

```
GROUP    webservers    My Web Server Group
```

Describing Security Groups

This section describes how to view currently configured security groups.

AWS Management Console

To view security groups

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Security Groups** in the **Navigation** pane.
The console displays a list of security groups that belong to the account.
3. To view more information about a security group, including its rules, select it.

Command Line Tools

To view security groups

- Enter the following command:

```
PROMPT> ec2-describe-group [group ...]
```

Amazon EC2 returns output similar to the following:

```
GROUP AIDADH4IGTRXXKCD webservers Web Servers
```

Adding a Security Group Rule

This section describes how to add a rule to a security group.

AWS Management Console

To add a rule to a security group

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Security Groups** in the **Navigation** pane.
The console displays a list of security groups that belong to the account.
3. Select a security group.
Its rules appear in the lower pane.
4. To add a rule, provide the following:

- **Protocol**

- **From Port**
 - **To Port**
5. Then, select from the following:
 - To allow access from other instances in a security group, enter the security group name in the Connection Source field.
 - To configure this rule to apply to an IP address range, enter the CIDR range in the Connection Source field. For example, enter 0.0.0.0/0 to allow all IP addresses to access the specified port range. Enter an IP address or subnet to limit access to that one computer or network, for example 92.23.32.51/32.
 6. Click **Save**.

The new rule is created and applied to all instances that belong to the security group.

Command Line Tools

To add a rule to a security group

- Enter the following command:

```
PROMPT> ec2-authorize group [-P protocol] (-p port_range | -  
t icmp_type_code) [-u source_group_user ...] [-o source_group ...] [-  
s source_subnet ...]
```

Amazon EC2 returns an elastic IP address similar to the following:

```
PERMISSION default ALLOWS tcp 22 22 FROM CIDR 126.52.1.130/32
```

Delete a Security Group Rule

This section describes how to delete a security group rule.

AWS Management Console

To delete a security group rule

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Security Groups** in the **Navigation** pane.

The console displays a list of security groups that belong to the account.
3. Select a security group.

Its rules appear in the lower pane.
4. To delete a rule, click its **Remove** button.

Amazon EC2 deletes the security group rule.

Command Line Tools

To delete a security group rule

- Enter the following command:

```
PROMPT> ec2-revoke  
group  
[-P protocol]
```

```
(-p port_range | -t icmp_type_code)
[-u source_group_user ...]
[-o source_group ...]
[-s source_subnet ...]
```

Amazon EC2 returns output similar to the following:

```
PERMISSION webservers ALLOWS tcp 80 80 FROM CIDR 205.192.0.0/16
```

Delete a Security Group

This section describes how to delete a security group.

AWS Management Console

To delete a security group

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Security Groups** in the **Navigation** pane.
The console displays a list of security groups that belong to the account.
3. Select a security group and click **Delete**.
A confirmation dialog box appears.
4. Click **Yes, Delete**.
Amazon EC2 deletes the security group.

Command Line Tools

To delete a security group

- Enter the following command:

```
PROMPT> ec2-delete-group group
```

Amazon EC2 returns output similar to the following:

```
GROUP webservers
```

Example

This section provides examples of configuring security groups using the command line tools.

Modifying the Default Group

This example shows Albert modifying the default group to meet his security needs.

Albert Modifies the Default Group

1	Albert launches a copy of his favorite public AMI . PROMPT> ec2-run-instances ami-eca54085 RESERVATION r-a034c7c9 924417782495 default INSTANCE i-cfd732a6 ami-eca54085 pending 0 m1.small 2007-07-11T16:40:44+0000 us-east-1c
---	--

2	<p>After a little wait for image launch to complete. Albert, who is a cautious type, checks the access rules of the default group.</p> <pre>PROMPT> ec2-describe-group default GROUP 598916040194 default default group PERMISSION default ALLOWS all FROM USER 598916040194 GRPNAME default</pre> <p>Albert notices that it only accepts ingress network connections from other members of the default group for all protocols and ports.</p>
3	<p>Albert, being paranoid as well as cautious, uses the Linux and UNIX <code>nmap</code> command to port scan his instance.</p> <pre>\$ nmap -P0 -p1-100 ec2-67-202-51-105.compute-1.amazonaws.com Starting nmap 3.81 (http://www.insecure.org/nmap/) at 2006-08-07 15:42 SAST All 100 scanned ports on ec2-67-202-51-105.compute-1.amazonaws.com (67.202.51.105) are: filtered Nmap finished: 1 IP address (1 host up) scanned in 31.008 seconds</pre>
4	<p>Albert decides he should be able to SSH into his instance, but only from his own machine.</p> <pre>PROMPT> ec2-authorize default -P tcp -p 22 -s 126.52.1.130/32 GROUP default PERMISSION default ALLOWS tcp 22 22 FROM CIDR 126.52.1.130/32</pre>
5	<p>Albert repeats the Linux and UNIX <code>nmap</code> port scan.</p> <pre>\$ nmap -P0 -p1-100 ec2-67-202-51-105.compute-1.amazonaws.com Starting nmap 3.81 (http://www.insecure.org/nmap/) at 2006-08-07 15:43 SAST Interesting ports on ec2-67-202-51-105.compute-1.amazonaws.com (67.202.51.105): (The 99 ports scanned but not shown are in state: filtered) PORT STATE SERVICE 22/tcp open ssh Nmap finished: 1 IP address (1 host up) scanned in 32.705 seconds Albert is happy (or at least less paranoid).</pre>

Creating a Three-Tier Web Service

Mary wants to deploy her public, failure resilient, three-tier web service (web, application, and database servers) in Amazon EC2. Her grand plan is to have her web tier start off executing in seven instances of `ami-fba54092`, her application tier executing in twenty instances of `ami-e3a5408a`, and her multi-master database in two instances of `ami-f1a54098`. She's concerned about the security of her subscriber database, so she wants to restrict network access to her middle and back tier machines. When the traffic to her site increases over the holiday shopping period, she adds additional instances to her web and application tiers to handle the extra load.

Launch Process

1	<p>First, Mary creates a group for her Apache web server instances and allows HTTP access to the world.</p> <pre>PROMPT> ec2-add-group apache -d "Mary's Apache group" GROUP apache Mary's Apache group PROMPT> ec2-describe-group apache GROUP 598916040194 apache Mary's Apache group PROMPT> ec2-authorize apache -P tcp -p 80 -s 0.0.0.0/0 GROUP apache PERMISSION apache ALLOWS tcp 80 80 FROM CIDR 0.0.0.0/0 PROMPT> ec2-describe-group apache GROUP 598916040194 apache Mary's Apache group PERMISSION 598916040194 apache ALLOWS tcp 80 80 FROM CIDR 0.0.0.0/0</pre>
2	<p>Mary launches seven instances of her web server AMI as members of the <code>apache</code> group.</p> <pre>PROMPT> ec2run ami-fba54092 -n 7 -g apache RESERVATION r-0592776c 598916040194 default INSTANCE i-cfd732a6 ami-fba54092 pending 0 m1.small 2007-07-11T16:40:44+0000 us-east-1c INSTANCE i-cfd732a7 ami-fba54092 pending 0 m1.small 2007-07-11T16:40:44+0000 us-east-1c INSTANCE i-cfd732a8 ami-fba54092 pending 0 m1.small 2007-07-11T16:40:44+0000 us-east-1c INSTANCE i-cfd732a9 ami-fba54092 pending 0 m1.small 2007-07-11T16:40:44+0000 us-east-1c INSTANCE i-cfd732aa ami-fba54092 pending 0 m1.small 2007-07-11T16:40:44+0000 us-east-1c INSTANCE i-cfd732ab ami-fba54092 pending 0 m1.small 2007-07-11T16:40:44+0000 us-east-1c INSTANCE i-cfd732ac ami-fba54092 pending 0 m1.small 2007-07-11T16:40:44+0000 us-east-1c PROMPT> ec2din i-cfd732a6 RESERVATION r-0592776c 598916040194 INSTANCE i-cfd732a6 ami-fba54092 ec2-67-202-51-245.compute-1.amazonaws.com running 0 m1.small 2007-07-11T16:40:44+0000</pre>

Amazon Elastic Compute Cloud User Guide
Example

3	<p>Being as paranoid as Albert, Mary uses the Linux and UNIX <code>nmap</code> command to confirm the permissions she just configured.</p> <pre>\$ nmap -P0 -p1-100 ec2-67-202-51-245.compute-1.amazonaws.com Starting nmap 3.81 (http://www.insecure.org/nmap/) at 2006-08-07 16:21 SAST Interesting ports on ec2-67-202-51-245.compute-1.amazonaws.com (67.202.51.245): (The 99 ports scanned but not shown are in state: filtered) PORT STATE SERVICE 80/tcp open http Nmap finished: 1 IP address (1 host up) scanned in 33.409 seconds</pre>
4	<p>Mary verifies her web server can be reached.</p> <pre>\$ telnet ec2-67-202-51-245.compute-1.amazonaws.com 80 Trying 67.202.51.245... Connected to ec2-67-202-51-245.compute-1.amazonaws.com (67.202.51.245). Escape character is '^]'. Mary can reach her web server.</pre>
5	<p>Mary creates a separate group for her application server.</p> <pre>PROMPT> ec2-add-group appserver -d "Mary's app server" GROUP appserver Mary's app server</pre>
6	<p>Mary starts twenty instances as members of <code>appserver</code> group.</p> <pre>PROMPT> ec2run ami-e3a5408a -n 20 -g appserver</pre>
7	<p>Mary grants network access between her web server group and the application server group.</p> <pre>PROMPT> ec2-authorize appserver -o apache -u AIDADH4IGTRXXKCD GROUP appserver PERMISSION appserver ALLOWS all FROM USER AIDADH4IGTRXXKCD GRPNAME apache</pre>
8	<p>Mary verifies access to her app server is restricted by port scanning one of the application servers using the Linux and UNIX <code>nmap</code> command.</p> <pre>\$ nmap -P0 -p1-100 ec2-67-202-51-162.compute-1.amazonaws.com Starting nmap 3.81 (http://www.insecure.org/nmap/) at 2006-08-07 15:42 SAST All 100 scanned ports on ec2-67-202-51-162.compute-1.amazonaws.com (67.202.51.162) are: filtered Nmap finished: 1 IP address (1 host up) scanned in 31.008 seconds</pre>

9	<p>Mary confirms that her web servers have access to her application servers.</p> <p>A. She (temporarily) grants SSH access from her workstation to the web server group:</p> <pre>PROMPT> ec2-authorize apache -P tcp -p 22 -s 126.52.1.130/32</pre> <p>B. She logs in to one of her web servers and connects to an application server on TCP port 8080.</p> <pre>\$ telnet ec2-67-202-51-162.compute-1.amazonaws.com 8080 Trying 67.202.51.162... Connected to ec2-67-202-51-162.compute-1.amazonaws.com (67.202.51.162). Escape character is '^]'</pre> <p>C. Satisfied with the setup, she revokes SSH access to the web server group.</p> <pre>PROMPT> ec2-revoke apache -P tcp -p 22 -s 126.52.1.130/32</pre>
10	<p>Mary repeats these steps to create the database server group and to grant access between the application server and database server groups.</p>

Related Topics

- [Network Security Concepts \(p. 18\)](#)

Using Regions and Availability Zones

Topics

- [API Overview](#) (p. 102)
- [Describing Regions and Availability Zones](#) (p. 102)
- [How to Configure Your Environment](#) (p. 103)
- [Launching Instances in Specific Availability Zones](#) (p. 103)
- [Related Topics](#) (p. 105)

This section describes how to work with regions and Availability Zones.



Note

Data transfer between regions is charged at the Internet data transfer rate for both the sending and the receiving region. For detailed information on Amazon EC2 charges, go to the [Amazon EC2 Product Page](#).

API Overview

This section provides a brief overview of each operation.

- **DescribeAvailabilityZones**—Describes Availability Zones available to your account
- **DescribeRegions**—Describes regions available to your account



Note

Although the region and Availability Zone operation list is limited, you can specify a region or Availability Zone for many operations.

For information about region and Availability Zone concepts, see [Availability Zones](#) (p. 20).

Describing Regions and Availability Zones

This section describes how to determine which regions and Availability Zones are available.

AWS Management Console

To find regions and Availability Zones

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. To determine available regions, select the **Region** list box in the **Navigation** pane.
After selecting a region, you can view Availability Zones within that region when launching an instance or creating a new Amazon EBS volume.
3. To view Availability Zones within a region, click **Volumes** in the **Navigation** pane.
The **Create Volume** dialog box appears.
4. To view Availability Zones within the region, select the **Availability Zones** list box.
5. When you are finished, click **Cancel**.

Command Line Tools

To describe regions and Availability Zones

1. Enter the following command to describe regions:

```
PROMPT> ec2-describe-regions
```

Amazon EC2 returns output similar to the following:

```
REGION          us-east-1          us-east-1.ec2.amazonaws.com
REGION          eu-west-1          eu-west-1.ec2.amazonaws.com
```

2. Enter the following command to describe Availability Zones within the `us-east-1` region:

```
PROMPT> ec2-describe-availability-zones --region eu-west-1
```

Amazon EC2 returns output similar to the following:

```
AVAILABILITYZONE us-east-1a    available    eu-west-1
AVAILABILITYZONE us-east-1b    available    eu-west-1
AVAILABILITYZONE us-east-1c    available    eu-west-1
AVAILABILITYZONE us-east-1d    available    eu-west-1
```

How to Configure Your Environment

After choosing a region and one or more Availability Zones, you should set up your environment.



Note

The `us-east-1.ec2.amazonaws.com` region is the original Amazon EC2 region and is selected by default.

To set up your environment

1. If you are using the command line tools, change the `EC2_URL` environment variable to point to the correct region.



Note

You can also use the `--region` command line option, or override the URL endpoint using the `-U` flag.

2. If you are using the AWS Management Console, you do not need to take any steps.
3. If you are using APIs, configure your application to use the appropriate service endpoint.

Launching Instances in Specific Availability Zones

When you launch an instance, you can optionally specify an Availability Zone. If you do not specify an Availability Zone, Amazon EC2 selects one for you in the region that you are using. When launching your initial instances, we recommend accepting the default Availability Zone, which allows Amazon EC2 to select the best Availability Zone for you based on system health and available capacity. Even if you have other instances running, you might consider not specifying an Availability Zone if your new instances do not need to be close to, or separated from, your existing instances.



Note

Availability Zones are not the same across accounts. The Availability Zone us-east-1a for account A is not necessarily the same as us-east-1a for account B. Zone assignments are mapped independently for each account.

You are charged a small bandwidth charge for data that crosses Availability Zones. For more information, go to the [Amazon EC2 product page](#).

AWS Management Console

To launch an instance in a specific Availability Zone

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Instances** in the **Navigation** pane.
The console displays a list of running instances.
3. Click **Launch Instance**.
The Launch Instance wizard appears.
4. Locate an AMI from the **Quick Start**, **My AMIs**, or **Community AMIs** tab and click **Select**.
The **Configure Firewall** page of the wizard appears.
5. If you have not configured a security group, complete the options on the **Configure Firewall** page. Otherwise, click **Skip**.
The **Launch** page of the wizard appears.
6. Confirm the following settings and click **Launch**.
 - AMI name
 - Number of instances to launch
 - Instance type to launch
 - Name of the key pair to launch the instance
 - Security group(s) within which to launch the instance

The instance(s) begin launching.

Command Line Tools

To launch an instance in a specific Availability Zone

1. Enter the following command to describe Availability Zones:

```
PROMPT> ec2-describe-availability-zones --region region
```

Amazon EC2 returns output similar to the following:

```
AVAILABILITYZONE    us-east-1a    available
AVAILABILITYZONE    us-east-1b    available
AVAILABILITYZONE    us-east-1c    available
AVAILABILITYZONE    us-east-1d    available
```

2. After determining the Availability Zones that are available to you, you can launch instances in any of them.

```
PROMPT> ec2-run-instances ami_id -n count --availability-zone zone
```

Amazon EC2 returns output similar to the following:

```
RESERVATION r-0ea54067 495219933132 default
INSTANCE i-3ea74257 ami-6ba54002 pending 0 m1.small
2007-07-11T16:40:44+0000 us-east-1a
INSTANCE i-31a74258 ami-6ba54002 pending 1 m1.small
2007-07-11T16:40:44+0000 us-east-1a
INSTANCE i-31a74259 ami-6ba54002 pending 2 m1.small
2007-07-11T16:40:44+0000 us-east-1a
INSTANCE i-31a7425a ami-6ba54002 pending 3 m1.small
2007-07-11T16:40:44+0000 us-east-1a
INSTANCE i-31a7425b ami-6ba54002 pending 4 m1.small
2007-07-11T16:40:44+0000 us-east-1a
INSTANCE i-31a7425c ami-6ba54002 pending 5 m1.small
2007-07-11T16:40:44+0000 us-east-1a
```

Related Topics

- [Availability Zones \(p. 20\)](#)
- [Region and Availability Zone FAQ \(p. 127\)](#)

Using Amazon Elastic Block Store

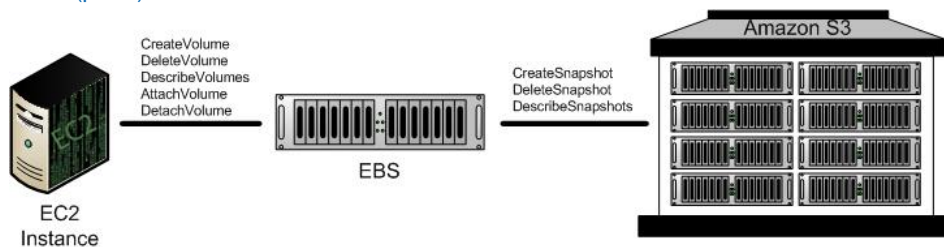
Topics

- [Amazon EBS API Overview](#) (p. 106)
- [How to Create an Amazon EBS Volume](#) (p. 107)
- [How to Attach the Volume to an Instance](#) (p. 107)
- [How to Describe Volumes and Instances](#) (p. 108)
- [How to Make an Amazon EBS Volume Available for Use](#) (p. 110)
- [How to Create an Amazon EBS Snapshot](#) (p. 110)
- [How to Describe Snapshots](#) (p. 111)
- [How to Detach an Amazon EBS Volume from an Instance](#) (p. 112)
- [How to Delete an Amazon EBS Snapshot](#) (p. 113)
- [How to Delete an Amazon EBS Volume](#) (p. 113)
- [Related Topics](#) (p. 114)

This section provides examples of how to create and use Amazon Elastic Block Store (Amazon EBS) volumes.

Amazon EBS API Overview

To configure and use Amazon EBS, we provide eight API functions. This section provides a brief overview of each function. For information on Amazon EBS concepts, see [Amazon Elastic Block Store](#) (p. 21).



- **CreateVolume**—Creates a new Amazon EBS volume using the specified size or creates a new volume based on a previously created snapshot.
- **DeleteVolume**—Deletes the specified volume.
This function does not delete any snapshots that were created from this volume.
- **DescribeVolumes**—Describes all volumes, including size, source snapshot, Availability Zone, creation time, and status (*available*, *in-use*).
- **AttachVolume**—Attaches the specified volume to a specified instance, exposing the volume using the specified device name.
A volume can only be attached to a single instance at any time. The volume and instance must be in the same Availability Zone and the instance must be running.
- **DetachVolume**—Detaches the specified volume from the instance to which it is attached.
This operation does not delete the volume. The volume can be attached to another instance and will have the same data as when it was detached.
- **CreateSnapshot**—Creates a snapshot of the volume you specify.
Once created, you can use the snapshot to create volumes that contain exactly the same data as the original volume.
- **DeleteSnapshot**—Deletes the specified snapshot.
This function does not affect currently running Amazon EBS volumes, regardless of whether they were used to create the snapshot or were derived from the snapshot.

- **DescribeSnapshots**—Describes all snapshots, including their source volume, snapshot initiation time, progress (percentage complete), and status (*pending*, *completed*).

How to Create an Amazon EBS Volume

To use Amazon EBS, you first create a volume that can be attached to any Amazon EC2 instance within the same Availability Zone. This example creates an 800 GiB Amazon EBS volume.

AWS Management Console

To create an Amazon EBS volume

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Volumes** in the **Navigation** pane.
The console displays a list of current volumes.
3. Click **Create Volume**.
The **Create Volume** dialog box appears.
4. Configure the following settings and click **Create**.
 - Size of the volume (in GiB)
 - Availability Zone in which to launch the instance
 - The ID of the snapshot from which you are launching the volume (optional)

Amazon EC2 begins creating the volume.

Command Line Tools

To create an Amazon EBS volume

1. Enter the following command.

```
PROMPT> ec2-create-volume --size 800 --zone us-east-1a
```

Amazon EBS returns information about the volume similar to the following example.

```
VOLUME vol-4d826724 800 us-east-1a available 2008-02-14T00:00:00+0000
```

2. To check whether the volume is ready, use the following command.

```
PROMPT> ec2-describe-volumes vol-4d826724
```

Amazon EBS returns information about the volume similar to the following example.

```
VOLUME vol-4d826724 800 us-east-1a available 2008-07-29T08:49:25+0000
```

How to Attach the Volume to an Instance

This section describes how to attach a volume that you created to an instance.



Note

Windows instances currently support devices xvda through xvdp. Devices xvda and xvdb are reserved by the operating system, xvdc is assigned to drive C:\, and, depending on the

instance type, devices xvdd through xvde might be reserved by the instance stores. Any device that is not reserved can be attached to an Amazon EBS volume. For a list of devices that are reserved by the instance stores, see [Instance Storage \(p. 13\)](#).

AWS Management Console

To attach an Amazon EBS volume

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Volumes** in the **Navigation** pane.
The console displays a list of current volumes.
3. Click **Create Volume**.
The **Create Volume** dialog box appears.
4. Select a volume and click **Attach Volume**.
The **Attach Volume** dialog box appears.
5. Select the instance to which the volume will attach from the **Instance** list box.
6. Select how the device is exposed to the instance from the **Device** list box.
7. Click **Attach**.
The volume is attached to the instance.

Command Line Tools

To attach an Amazon EBS volume

- Enter the following command.

```
PROMPT> ec2-attach-volume volume_id -i instance_id -d device
```

Amazon EBS returns information similar to the following.

```
ATTACHMENT volume_id instance_id device attaching date_time
```

Example

This example attaches volume `vol-4d826724` to instance `i-6058a509` in Linux and UNIX and exposes it as device `/dev/sdh`.

```
PROMPT> ec2-attach-volume vol-4d826724 -i i-6058a509 -d /dev/sdh
```

```
ATTACHMENT vol-4d826724 i-6058a509 /dev/sdh attaching  
2008-02-14T00:15:00+0000
```

This example attaches volume `vol-4d826724` to instance `i-6058a509` in Windows and exposes it as device `xvdf`.

```
PROMPT> ec2-attach-volume vol-4d826724 -i i-6058a509 -d xvdf
```

```
ATTACHMENT vol-4d826724 i-6058a509 xvdf attaching 2008-02-14T00:15:00+0000
```

How to Describe Volumes and Instances

After creating Amazon EBS volumes and attaching them to instances, you should verify they are available.

AWS Management Console

To view information about an Amazon EBS volume

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Volumes** in the **Navigation** pane.
The console displays a list of current volumes and the instances to which they are attached.
3. To view more information about a volume, select it.
Information about the volume appears in the lower pane.

Command Line Tools

To describe volumes, which lists information about all volumes that you own

- Enter the following command.

```
PROMPT> ec2-describe-volumes
```

Amazon EBS returns the following information.

```
VOLUME vol-4d826724 us-east-1a 800 in-use 2008-02-14T00:00:00+0000
ATTACHMENT vol-4d826724 i-6058a509 /dev/sdh attached
2008-02-14T00:00:17+0000
VOLUME vol-50957039 13 us-east-1a available 2008-02-09T00:00:00+0000
VOLUME vol-6682670f 1 us-east-1a in-use 2008-02-11T12:00:00+0000
ATTACHMENT vol-6682670f i-69a54000 /dev/sdh attached
2008-02-11T13:56:00+0000
```

This information includes the volume ID, capacity, status (in-use or available) and creation time of each volume. If the volume is attached, an attachment line shows the volume ID, the instance ID to which the volume is attached, the device name exposed to the instance, its status (attaching, attached, detaching, detached), and when it attached.

To describe instances, which lists volumes that are attached to running instances

- Enter the following command.

```
PROMPT> ec2-describe-instances
```

Amazon EBS returns the following information.

```
RESERVATION r-e112fc88 416161254515 default
INSTANCE i-3b887c52 ami-3fd13456
ec2-67-202-27-216.compute-1.amazonaws.com
domU-12-31-38-00-35-94.compute-1.internal
running gsg-keypair 0 m1.small 2007-11-26T13:20:35+0000 windows
vol-4d826724
RESERVATION r-e612fc8f 416161254515 default
INSTANCE i-21b63c22 ami-3fd13456
ec2-67-202-18-227.compute-1.amazonaws.com
domU-12-31-38-00-39-28.compute-1.internal
running gsg-keypair 0 m1.small 2007-11-26T13:21:51+0000 windows
vol-6682670f
```

How to Make an Amazon EBS Volume Available for Use

Inside the instance, the Amazon EBS volume is exposed as a normal block device and can be formatted as any file system and mounted.

After making the Amazon EBS volume available for use, you can take snapshots of it for backup purposes or to use as baselines to launch new volumes.

Linux and UNIX

This section describes how to make a volume available to the Linux and UNIX operating system.

To create an ext3 file system on the Amazon EBS volume and mount it as /mnt/data-store

1. Enter the following command.

```
$ yes | mkfs -t ext3 /dev/sdh
```

2. Enter the following command.

```
$ mkdir /mnt/data-store
```

3. Enter the following command.

```
$ mount /dev/sdh /mnt/data-store
```

Any data written to this file system is written to the Amazon EBS volume and is transparent to applications using the device.

Windows

This section describes how to make a volume available to the Windows operating system.

To use an Amazon EBS volume

1. Log in to your instance using Remote Desktop.
2. On the taskbar, click **Start**, and then click **Run**.
3. Type **diskmgmt.msc** and click **OK**. The Disk Management utility opens.
4. Right-click the Amazon EBS volume, select **New Volume**, and follow the on-screen prompts.



Note

If the **New Volume** option does not appear, select **Format**.

Any data written to this file system is written to the Amazon EBS volume and is transparent to applications using the device.

How to Create an Amazon EBS Snapshot

After writing data to an Amazon EBS volume, you can periodically create a snapshot of the volume to use as a baseline for new volumes or for data backup.



Note

Snapshots occur asynchronously and the status settings of volumes indicate "pending" until they complete.

For information on creating an Amazon EBS volume from a snapshot, see [Amazon Elastic Block Store \(p. 21\)](#).

AWS Management Console

To create a snapshot

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Snapshots** in the **Navigation** pane.
The console displays a list of current snapshots.
3. Click **Create Snapshot**.
The Create Snapshot dialog box appears.
4. Select the volume to create a snapshot for and click **Create**.
Amazon EC2 begins creating the snapshot.

Command Line Tools

To create a snapshot

- Enter the following command.

```
PROMPT> ec2-create-snapshot volume_id
```

Amazon EBS returns information similar to the following example.

```
SNAPSHOT snap-78a54011 vol-4d826724 pending 2008-02-15T09:03:58+0000
```

How to Describe Snapshots

This section describes how to view snapshots that you created.



Note

When the snapshot is complete, its status will change to `completed` and the percentage will change to 100%.

AWS Management Console

To describe snapshots

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Snapshots** in the **Navigation** pane.
The console displays a list of current snapshots and their status.
3. To view more information about a snapshot, select it.
Information about the snapshot appears in the lower pane.

Command Line Tools

To describe snapshots

- Enter the following command.

```
PROMPT> ec2-describe-snapshots snap-78a54011
```

Amazon EBS returns information about all snapshots that you own.

```
SNAPSHOT snap-78a54011 vol-4d826724 pending 2008-02-15T09:03:58+0000 60%
```

How to Detach an Amazon EBS Volume from an Instance

An Amazon EBS volume can be detached from an instance by either explicitly detaching the volume or terminating the instance. This example unmounts the volume and explicitly detaches it from the instance. This is useful when you want to terminate an instance or attach a volume to a different instance.



Caution

A volume must be unmounted inside the instance before being detached. Failure to do so will result in damage to the file system or the data it contains.

To verify the volume is no longer attached to the instance, see [How to Describe Snapshots \(p. 111\)](#).

AWS Management Console

To detach an Amazon EBS volume

1. Enter the following command from the command line.

```
# umount -d /dev/sdh
```
2. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
3. Click **Volumes** in the **Navigation** pane.

The console displays a list of current volumes.
4. Select a volume and click **Detach Volume**.

A confirmation dialog box appears.
5. Click **Yes, Detach**.

The volume is detached from the instance.

Command Line Tools

To detach an Amazon EBS volume

- Enter the following commands.

```
# umount -d /dev/sdh  
PROMPT> ec2-detach-volume vol-4d826724
```

Amazon EBS returns information similar to the following example.

```
ATTACHMENT vol-4d826724 i-6058a509 /dev/sdh detaching  
2008-02-14T00:00:17+0000
```

To detach an Amazon EBS volume by terminating the instance

- Enter the following command.

```
PROMPT> ec2-terminate-instances i-6058a509
```

Amazon EBS returns information similar to the following example.

```
INSTANCE    i-6058a509    running shutting-down
```

How to Delete an Amazon EBS Snapshot

After a snapshot is no longer needed, it can be deleted. This section describes how to delete a snapshot.

AWS Management Console

To delete a snapshot

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Snapshots** in the **Navigation** pane.
The console displays a list of current snapshots.
3. Select a snapshot and click **Delete Snapshot**.
A confirmation dialog box appears.
4. Click **Yes, Delete**.
The snapshot is deleted.

Command Line Tools

To delete a snapshot

- Enter the following command.

```
PROMPT> ec2-delete-snapshot snapshot_id
```

Amazon EBS returns information similar to the following example.

```
SNAPSHOT snap-78a54011
```

How to Delete an Amazon EBS Volume

After a volume is no longer needed, it can be deleted. Once deleted, its data is deleted and it cannot be attached to any instance. However, you can store a snapshot of the volume that you can use to recreate it later.

This section describes how to delete a volume.

AWS Management Console

To delete a volume

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Volumes** in the **Navigation** pane.
The console displays a list of current volumes.
3. Select a volume and click **Delete Volume**.
A confirmation dialog box appears.
4. Click **Yes, Delete**.
The volume is deleted.

Command Line Tools

To delete a volume

- Enter the following command.

```
PROMPT> ec2-delete-volume vol-4282672b
```

Amazon EBS returns information similar to the following example.

```
VOLUME vol-4282672b
```

Related Topics

- [Amazon Elastic Block Store \(p. 21\)](#)

Using Auto Scaling, Elastic Load Balancing, and Amazon CloudWatch

Topics

- [Auto Scaling \(p. 115\)](#)
- [Elastic Load Balancing \(p. 115\)](#)
- [Amazon CloudWatch \(p. 115\)](#)

This section describes how to get started with Amazon EC2 fault resilient features.

Auto Scaling

Auto Scaling enables you to scale up or down the number of instances you are using based on parameters that you specify, such as traffic or CPU load.

For information on setting up Auto Scaling, refer to the *Amazon Auto Scaling Developer Guide*.

Elastic Load Balancing

Elastic Load Balancing lets you automatically distribute the incoming traffic (or load) among all the instances you are running. The service also makes it easy to add new instances when you need to increase the capacity of your web site application.

For information on setting up Elastic Load Balancing, refer to the *Elastic Load Balancing Developer Guide*.

Amazon CloudWatch

Amazon CloudWatch collects raw data from partnered AWS services such as Amazon EC2 and then processes the information into readable, near real-time metrics. These statistics are recorded for a period of two weeks, allowing you access to historical information and providing you with a better perspective on how your web application or service is performing.

This section describes how to enable monitoring on a new instance and how to enable monitoring on a running instance. For detailed information about Amazon CloudWatch, refer to the *Amazon CloudWatch Developer Guide*.

Enabling Amazon CloudWatch on a New Amazon EC2 Instance

This section describes how to enable monitoring when running a new instance.

Command Line Tools

To launch an instance

1. Use the `ec2-run-instances` command.

```
PROMPT> ec2-run-instances ami-235fba4a -k gsg-keypair --monitoring
```

Amazon EC2 returns output similar to the following example.

```
RESERVATION    r-7430c31d    924417782495    default
```

```
INSTANCE          i-ae0bf0c7          ami-2bb65342      pending gsg-keypair  0
m1.small          2008-03-21T16:19:25+0000  us-east-1a      monitoring-
enabled
```

2. Verify that monitoring is enabled for the instance. The instance starts running in a few minutes.
3. The following command displays the launch status of the instance.

```
PROMPT> ec2-describe-instances i-ae0bf0c7
RESERVATION      r-7430c31d          924417782495      default
INSTANCE         i-ae0bf0c7          ami-2bb65342
ec2-67-202-7-236.compute-1.amazonaws.com
ip-10-251-31-162.ec2.internal  running gsg-keypair  0
m1.small         2008-03-21T16:19:25+0000us-east-1a  monitoring-enabled
```



Important

After launching an instance, you are billed hourly for running time. When you are finished, make sure to terminate any instances that you started.

Enabling Amazon CloudWatch on an Existing Amazon EC2 Instance

This section describes how to enable monitoring for a running instance.

Command Line Tools

To monitor an instance

1. Use the `ec2-monitor-instances` command.

```
PROMPT> ec2-monitor-instances instance_id
```

Amazon EC2 returns output similar to the following example.

```
i-ae0bf0c7  monitoring-pending
```

2. The following command displays the monitoring status of the instance.

```
PROMPT> ec2-describe-instances instance_id
RESERVATION      r-7430c31d          924417782495      default
INSTANCE         i-ae0bf0c7          ami-2bb65342
ec2-67-202-7-236.compute-1.amazonaws.com
ip-10-251-31-162.ec2.internal  running gsg-keypair  0
m1.small         2008-03-21T16:19:25+0000us-east-1a  monitoring-enabled
```


Using Public Data Sets

Topics

- [Finding Public Data Sets \(p. 117\)](#)
- [Launching an Instance \(p. 117\)](#)
- [Launching a Public Data Set Volume \(p. 117\)](#)
- [Mounting the Public Data Set Volume \(p. 118\)](#)
- [Related Topics \(p. 118\)](#)

This section describes how to use Amazon EC2 public data sets.

Finding Public Data Sets

Before you launch a public data set, you must locate the set to launch.

To find a public data set

1. Go to the [Public Data Sets Page](#).
2. Locate a public data set and write down its snapshot ID for your operating platform (e.g., Windows, Linux/UNIX).

Launching an Instance

Launch an instance as you normally do. For more information, see [Launching and Using Instances \(p. 64\)](#).

Launching a Public Data Set Volume

To use a public data set, you launch an Amazon EBS volume, specifying its snapshot ID.

AWS Management Console

To create an Amazon EBS volume

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Volumes** in the **Navigation** pane.
The console displays a list of current volumes.
3. Click **Create Volume**.
The **Create Volume** dialog box appears.
4. Configure the following settings and click **Create**.
 - Size of the volume (in GiB) (optional)
 - Availability Zone in which to launch the instance
 - The ID of the public data set snapshot

Amazon EC2 begins creating the volume.

Command Line Tools

To create an Amazon EBS volume

1. Enter the following command.

```
PROMPT> ec2-create-volume --snapshot public-data-set-snapshot-id --  
zone availability-zone
```

Amazon EBS returns information about the volume similar to the following example.

```
VOLUME vol-4d826724 85 us-east-1a available 2008-02-14T00:00:00+0000
```

2. To check whether the volume is ready, use the following command.

```
PROMPT> ec2-describe-volumes vol-4d826724
```

Amazon EBS returns information about the volume similar to the following example.

```
VOLUME vol-4d826724 85 us-east-1a available 2008-07-29T08:49:25+0000
```

Mounting the Public Data Set Volume

Mount the volume as you normally do. For more information, see [How to Make an Amazon EBS Volume Available for Use](#) (p. 110).

Related Topics

- [Public Data Set Concepts](#) (p. 25)

Reserving Amazon EC2 Instances

Topics

- [How to Find and Purchase Reserved Instances](#) (p. 119)
- [Related Topics](#) (p. 120)

This section describes how to find and purchase available Amazon EC2 Reserved Instances.

How to Find and Purchase Reserved Instances

This section describes how to find and purchase Reserved Instances.

AWS Management Console

To find and purchase a Reserved Instance

1. Log in to the [AWS Management Console](#) and click the **Amazon EC2** tab.
2. Click **Instances** in the **Navigation** pane.
The console displays a list of running instances.
3. Select **Purchase Reserved Instance** from the **Reserve** list box.
4. Locate a Reserved Instance to purchase by specifying the following:
 - Platform
 - Instance Type
 - Availability Zone
 - Term

The AWS Management Console displays the **Fixed Reservation Price** and the **Usage Price**.

5. Select the number of instances to purchase and click **Continue**.
The AWS Management Console prompts you to confirm your order.
6. To confirm your order, click **Place Order**.
Your purchase is complete.
7. To verify your order, select **View Reserved Instance** from the **Reserve** list box.
The AWS Management Console displays a list of Reserved Instances that belong to your account.

Command Line Tools

To find and purchase a Reserved Instance

1. Check which Reserved Instances are available and where they are located:

```
PROMPT> ec2-describe-reserved-instances-offerings
```

Amazon EC2 returns output similar to the following:

```
PROMPT> ec2-describe-reserved-instances-offerings
OFFERING 4b2293b4-5813-4cc8-9ce3-1957fc1dcfc8 m1.small us-east-1a 3y 0.00
0.00 m1.small offering in us-east-1a
OFFERING 4b2293b4-5813-4cc8-9ce3-1957fc1dcfc7 c1.medium us-east-1b 1y 0.00
0.00 c1.medium offering in us-east-1b
```

```
OFFERING 4b2293b4-5813-4cc8-9ce3-1957fc1dcfc6 c1.xlarge us-east-1c 1y 0.00  
0.00 c1.xlarge offering in us-east-1c
```

In this example, there are 3 year Reserved Instances available for m1.small instances in us-east-1a, 1 year Reserved Instance available for c1.medium instances in us-east-1b, and 1 year Reserved Instances available for c1.xlarge instances in us-east-1a.

2. After you know what Reserved Instances are available, you can purchase them. To purchase a Reserved Instance:

```
PROMPT> ec2-purchase-reserved-instances-offering --offering offering_id --  
instance-count count
```

Amazon EC2 returns output similar to the following:

```
PURCHASE 4b2293b4-5813-4cc8-9ce3-1957fc1dcfc8 af9f760e-  
c1c1-449b-8128-1342d3a6927a af9f760e-c1c1-449b-8128-1342d3a6927b  
af9f760e-c1c1-449b-8128-1342d3a6927c
```

3. Write down or save the reservation numbers for future reference.
4. Verify the purchase:

```
PROMPT> ec2-describe-reserved-instances
```

Amazon EC2 returns output similar to the following:

```
RESERVEDINSTANCE af9f760e-c1c1-449b-8128-1342d3a6927a m1.small us-east-1a  
12 0.00 0.00 19 m1.small offering in us-east-1a Active  
4b2293b4-5813-4cc8-9ce3-1957fc1dcfc8 m1.small us-east-1a 12 0.00 0.00 3  
m1.small offering in us-east-1a Active  
RESERVEDINSTANCE af9f760e-c1c1-449b-8128-1342d3a6927b m1.small us-east-1a  
12 0.00 0.00 19 m1.small offering in us-east-1a Active  
4b2293b4-5813-4cc8-9ce3-1957fc1dcfc8 m1.small us-east-1a 12 0.00 0.00 3  
m1.small offering in us-east-1a Active  
RESERVEDINSTANCE af9f760e-c1c1-449b-8128-1342d3a6927c m1.small us-east-1a  
12 0.00 0.00 19 m1.small offering in us-east-1a Active  
4b2293b4-5813-4cc8-9ce3-1957fc1dcfc8 m1.small us-east-1a 12 0.00 0.00 3  
m1.small offering in us-east-1a Active
```

Related Topics

- [Reserved Instance Concepts \(p. 14\)](#)
- [Reserved Instances FAQs \(p. 130\)](#)

Technical FAQ

Topics

- [General Information FAQ \(p. 121\)](#)
- [Operation Information FAQ \(p. 122\)](#)
- [Instance Types and Architectures FAQ \(p. 123\)](#)
- [IP Information FAQ \(p. 125\)](#)
- [Region and Availability Zone FAQ \(p. 127\)](#)
- [Windows Instances FAQ \(p. 129\)](#)
- [Monitoring, Errors, and Unexpected Behavior FAQ \(p. 129\)](#)
- [Reserved Instances FAQs \(p. 130\)](#)
- [Paid AMIs FAQ \(p. 131\)](#)
- [Kernels, RAM Disks, and Block Device Mappings FAQ \(p. 133\)](#)
- [Error Messages FAQ \(p. 133\)](#)
- [Miscellaneous FAQ \(p. 134\)](#)

This section contains answers to commonly asked questions.

General Information FAQ

How many instances can I launch?

Each user has a concurrent running instance limit. For new users, this limit is 20. If you need more than 20 instances, please complete the [Amazon EC2 Instance Request Form](#) and your request will be considered.

How do I sign a request?

Information on signing SOAP requests is provided in the [Amazon Elastic Compute Cloud Developer Guide](#). Information on signing Query requests is provided in the [Amazon Elastic Compute Cloud Developer Guide](#).

What username do I use for the various Amazon EC2 tools?

When you sign up with Amazon Web Services, you are provided an AWS Account ID. This is your username. For more information, refer to the [Amazon Elastic Compute Cloud Getting Started Guide](#).

Why do my instances take so long to start?

Amazon EC2 must move the images around the network before they can be launched. For big images and/or congested networks, this can take several minutes. To improve performance, images are cached. As you launch your images more frequently, it should be less noticeable.

How durable are the instance stores?

Instance stores appear to an instance as a local disk. They will survive intentional and unintentional reboots of the instance unless the instance terminates or the underlying drive fails.

You should always backup or replicate important data.

What happens to my running instances if the machines on which they are running go down?

The instances will terminate and will need to be relaunched. The data on the instances' hard drives will be lost.

Always replicate important data or store it in Amazon S3.

Can I get a bigger/smaller/differently optimized virtual machine?

Yes. For more information, see [Instance Types \(p. 12\)](#).

Is there a REST interface to Amazon EC2?

Not at present. You can use the SOAP API, Query API, or the command line tools.

How does Amazon EC2 handle load balancing?

With a service as flexible as Amazon EC2, you can use many types of load balancing systems. The load balancing instances can forward traffic to other systems. There are several open source solutions that are in wide use.

Does Amazon perform system maintenance?

Yes. Periodically, Amazon might perform maintenance that requires a reboot of your system. Make sure your instances can recover and restart after being rebooted.

Operation Information FAQ

How do I handle time synchronization between instances?

You can set up NTP (Network Time Protocol). For more information, go to www.ntp.org. NTP is particularly important if you plan on using any Amazon web services (such as Amazon S3 or Amazon EC2) from within an instance, because requests to these services must be timestamped.

Is there a method for an instance to discover its own instance ID?

From within your instance you can use REST-like queries to <http://169.254.169.254/2009-04-04/> to retrieve various instance-specific metadata, including the instance ID. For more information, see [Instance Metadata \(p. 71\)](#).

Can I pass arbitrary configuration values to an instance at launch time?

Yes, although the size of the data is limited to 16K. For more information, see [Instance Metadata \(p. 71\)](#).

Is there a way to run a script on instance termination?

Not with any reliability. Amazon EC2 tries to shut an instance down cleanly (running system shutdown scripts), but there is only a short time available. In some cases (e.g., hardware failure), this does not happen.

Because there is no way to ensure shutdown scripts run, have a strategy to deal with abnormal terminations.

How can I allow other people to launch my AMIs?

You can allow other users to launch your AMIs by modifying the AMI's `launchPermission` attribute. You can grant public launch permissions or explicit permissions to specific users. For more information, see [How to Share AMIs \(p. 49\)](#).

Why do I need to reregister a rebundled AMI? Can I keep the same AMI ID?

An AMI ID is associated with the physical bits in an image. To protect users from images being modified, we require you to reregister AMIs after rebundling.

Can I pass JVM properties to the command line tools?

Yes. By setting the environment variable `EC2_JVM_ARGS`, you can pass arbitrary JVM properties to the command line tools.

Can I use a proxy with the command line tools?

Yes. By passing in JVM properties through the `EC2_JVM_ARGS` environment variable, you can specify proxy settings for the command line tools. For example, in Linux and UNIX:

```
export EC2_JVM_ARGS="-Dhttp.proxyHost=http://my.proxy.com -  
Dhttp.proxyPort=8080"
```

Properties for configuring a proxy are described in the following table.

Setting	Description
<code>https.proxyHost</code>	HTTPS proxy host
<code>https.proxyPort</code>	HTTPS proxy port
<code>http.proxyHost</code>	HTTP proxy host
<code>http.proxyPort</code>	HTTP proxy port
<code>http.proxyRealm</code>	Proxy realm (https and http)
<code>http.proxyUser</code>	Proxy username (https and http)
<code>http.proxyPass</code>	Proxy password (https and http)



Note

`https.proxyHost` should be used when `EC2_URL` points to an https host, and `http.proxyHost` when `EC2_URL` points to an http host.

Instance Types and Architectures FAQ

What happened to the original instance type?

The original instance type is still available. It is called the small instance (m1.small) and it has the same technical specifications.

Will the original instance type be retired soon?

There are no plans to retire the original instance type.

If I do not specify an instance type at launch, what type of instance will I get?

You will get a m1.small Amazon EC2 instance type.

Does my instance limit apply to all instance types or is there a separate limit for each type?

The instance limit applies to the sum of all instances, regardless of type. There is no separate instance limit per type.

Can I mix instance types, or do I have to use the same type for all of my instances?

You can launch any combination of instance types. Choose the instance types that have the most appropriate memory, CPU, and storage for each function within your application.

How do I select the right instance type?

Amazon EC2 instances are grouped into two families: standard and High-CPU. Standard instances have memory to CPU ratios suitable for most general purpose applications; High-CPU instances have proportionally more CPU resources than memory (RAM) and are well suited for compute-intensive applications. When selecting instance types, you might want to use less powerful instance types for your web server instances and more powerful instance types for your database instances. Additionally, you might want to run CPU instance types for CPU-intensive data processing tasks.

For most applications, the standard instance types are appropriate. These instance types include the small instance (m1.small), large instance (m1.large), and extra large instance (m1.xlarge). High-CPU instances are well suited for compute-intensive applications such as rendering, search indexing, and computational analysis. The High-CPU instance types are the High-CPU medium instance (c1.medium) and the High-CPU extra large instance (c1.xlarge). For more information, refer to [Instance Types \(p. 12\)](#).

One of the advantages of Amazon EC2 is that you pay by the instance hour, which makes it convenient and inexpensive to test the performance of your application on different instance families and types. One good way to determine the most appropriate instance family and instance type is to launch test instances and benchmark your application.

When should I use High-CPU instance types (c1.medium and c1.xlarge)?

High-CPU instance types have a proportionately higher ratio of CPU to memory and are well suited for compute-intensive applications. To determine whether they are appropriate for you, launch an instance and benchmark your own application on different instance types and calculate which is most appropriate.

Which instance types are 32-bit and which are 64-bit?

The small (m1.small) and High-CPU medium (c1.medium) instances are 32-bit. The large (m1.large), extra large (m1.xlarge), and High-CPU extra large (c1.xlarge) instances are 64-bit.

Can I launch any AMI on any type of instance?

No. You must use 64-bit AMIs on large (m1.large), extra large (m1.xlarge) and High-CPU extra large (c1.xlarge) instances. You must use 32-bit AMIs on small (m1.small) and High-CPU medium (c1.medium) instances.

Can I use my own kernel?

Not at present. However, as of version 2008-02-01 of the Amazon EC2 API you can use any of the kernels published by Amazon EC2 or selected vendors.

Do I have to do anything special to bundle the large or extra large instances?

Make sure to use the latest AMI Tools.

Can I build an AMI that works on both 32-bit and 64-bit instances?

No, an AMI is either a 32-bit AMI or a 64-bit.

Can I run 32-bit applications on 64-bit AMIs?

You can run a 32-bit application on a 64-bit host if the Linux/UNIX kernel is compiled with IA32 emulation and the correct 32-bit libraries are available.

By default, the Amazon DomU Kernel has IA32 emulation enabled and there are many public AMIs that include pre-installed 32-bit libraries. If the library you require is not included with the AMI, you can install it using standard tools (e.g., yum).

How fast is the disk?

The large and extra large instances have higher and more consistent I/O performance than the original (small) instance.



Note

The first write to any given block of the disk will be slower than subsequent writes. For more information, see [Disk Performance Optimization \(p. 76\)](#)

Can I RAID the spindles exposed on large and extra large instances?

Yes, you can use software RAID on top of the exposed spindles.



Note

The initial RAID setup might take a long time. For more information, see [Disk Performance Optimization \(p. 76\)](#)

IP Information FAQ

How do I host a public domain if I have to DHCP an IP address?

You can use a dynamic DNS service, such as [DynDNS](#) or [ZoneEdit](#). Alternatively, you can map an elastic IP address to your instance and avoid the propagation delays possible with a dynamic DNS solution.

Why do I get an internal (RFC 1918) IP address when I look up a DNS name that I expect to map to my instance's external IP address?

The Amazon EC2 DNS servers return the internal IP address when asked about an instance's public DNS name. In this way, DNS lookups that would resolve to a public Amazon EC2 IP address will be translated to the correct internal IP address. This only works when using the Amazon EC2 DNS servers from an Amazon EC2 instance.

Why is Amazon EC2 Using NAT?

Public IP space is a limited resource. Amazon EC2 is adopting NAT to ensure that we are able to efficiently make use of our public Internet addresses.

Furthermore, the new NAT networking will enable Amazon to deliver new features in the future. For example, some users might not want external addresses. This would allow for non-Internet routable clusters, which will further preserve IPs and increase security for those not running public facing servers.

Can I use a static IP in my instances?

Not at present. Your image must be configured as a DHCP client and it will be assigned an IP address. Currently, all instances come with Internet- addressable IP addresses. If you enable access through the firewall from the "world", you can address them from anywhere.

How does the instance know its public and private addresses?

From within the instance, issue the following HTTP queries:

To obtain the internal IP address:

```
curl http://169.254.169.254/2009-04-04//meta-data/local-ipv4
```

To obtain the public IP address:

```
curl http://169.254.169.254/2009-04-04//meta-data/public-ipv4
```

Why am I limited to 5 elastic IP addresses?

Public (IPv4) Internet addresses are a scarce public resource. Amazon EC2 is committed to helping use that space efficiently.

By default, all accounts are limited to 5 elastic IP addresses. If you need more than 5 Elastic IP addresses, please complete the [Amazon EC2 Elastic IP Address Request Form](#). We will ask you to think through your use case and help us understand your need for additional addresses.

Is my elastic IP addressed fixed to a single instance?

Unlike a traditional dedicated IP addresses, an elastic IP can be assigned to many different instances over time.

Is there a minimum usage required for elastic IP addresses?

When operating within the 5 address limit, you can leave addresses unattached as you need. However, we reserve the right to reclaim elastic IP addresses that are chronically underutilized.

Is there a charge for elastic IP addresses?

To ensure our customers are efficiently using elastic IP addresses, we impose the a small hourly charge when these IP addresses are not mapped to an instance. When these IP addresses are mapped to an instance, they are free of charge. To avoid charges for elastic IP addresses that you are not using, use `ReleaseAddress`.

Do I need one elastic IP address for every instance that I have running?

You do not need an elastic IP address for all your instances. By default, every instance comes with a private IP address and an Internet routable public IP address. These addresses are fixed for the life of the instance. We believe this should be adequate for many applications where you do not need a long lived Internet routable end point (e.g., compute clusters, web crawling, and backend services).

Why don't you use IPV6 addresses?

Because of the scarcity of IPV4 Internet address, Amazon EC2 will be actively investigating the use of IPV6 addresses. We believe this is the only tenable long term solution. We don't yet have a timeline for introducing IPV6 addresses, but when we do support IPV6 addresses, we will be able to remove the friction we have imposed with IPV4 address.

Can I launch an instance with no public IP address?

You cannot currently launch an instance without a public IP address. We understand that for many applications, it is desirable to have no Internet routable IP address (e.g., internal databases).

How long does it take to remap an elastic IP address?

After you successfully make an API call to remap an IP address, it will usually occur within a few minutes.

Will I be charged for the time when my IP address is unattached because my instance failed?

You are not charged until your elastic IP address has been unattached for a full hour. As long as you are monitoring your instances, you will have plenty of time to reattach your instance before the charge is metered.

Am I limited to 100 elastic IP remaps per month?

No. The first 100 remaps per account are free. After that, there will be a charge for each remap.

Region and Availability Zone FAQ

Why aren't regions tightly integrated with each other?

We isolate the regions from each other to achieve greater fault tolerance, improve stability, and to help prevent issues within one region from affecting another. To simplify using instances across regions, we provide tools such as `ec2-migrate-image` and `ec2-migrate-manifest`.

How do I interact with EC2 in different regions?

Use the region-specific service endpoint for the region you want. To get a list of regions and their endpoints, use the `DescribeRegions` API, for example:

```
PROMPT> ec2-describe-regions
```

```
REGION          us-east-1          us-east-1.ec2.amazonaws.com
REGION          eu-west-1          eu-west-1.ec2.amazonaws.com
```

How do I launch an AMI in another region?

Simply copy your AMI from its current bucket to a bucket located in the region where you want to launch the AMI and register the AMI. For example, to launch a US-based AMI in Europe, you have to copy the AMI to an Amazon S3 bucket that was created with an EU location constraint. After the AMI is copied, you must register the AMI and use the obtained AMI ID for launches in the new region.

Also, make sure to give read access to the bucket, image manifest, and image parts to `ec2-bundled-images@amazon.com` for Windows AMIs, and `za-team@amazon.com` for Linux AMIs.

What tools are available to help migrate my AMIs to a new region?

The API Tools contain a new command called `ec2-migrate-image`. It is designed to help migrate AMIs to a new region. Run `ec2-migrate-image --help` for more details.

Can I use the same SSH key pair across regions?

No. You must create a separate SSH key pair for each region.



Note

This is the key pair used for SSH connections to the instance. Your AWS Account ID credentials are global and you use them for all regions.

How do I launch an Amazon EBS volume from a snapshot across regions?

At this time, snapshots cannot be copied across regions. However, data on Amazon EBS volumes can be copied across regions out of band. For example, you can run an instance in the region with the source volume, run an instance in the destination region with a new volume attached, and use rsync or some other file copy mechanism to copy data.

If I make service calls to the `ec2.amazonaws.com` service endpoint, where will my instances launch?

They will launch in the original Amazon EC2 `us-east-1.ec2.amazonaws.com` region.

Can instances use group-based firewall rules across regions?

No. Group-based firewall rules only work within a region. If you need instances to communicate with each other across regions, you should use CIDR based firewall rules. To simplify IP address management, you can use firewall rules in combination with Elastic IP addresses.



Note

Because inter-region traffic crosses the public Internet, encrypt all sensitive data.

How do I use the command line tools with multiple regions?

By default, the command line tools use the original `us-east-1.ec2.amazonaws.com` region. To specify a different region, see [Region and Availability Zone Concepts \(p. 19\)](#).

What is the cost for data transfer between regions?

Data transferred from one region to another is charged at both sides at the Internet data transfer rate.

Can I assume that my Availability Zone `us-east-1a` is the same location as someone else's Availability Zone `us-east-1a`?

No. Currently, we do not support cross-account proximity. Each account's availability zones are independent. For example, the `us-east-1a` Availability Zone for one account might be in a different location than for another account.

How can I make sure that I am in the same Availability Zone as another developer?

We do not currently support the ability to coordinate availability groups between developer accounts. We are seeking customer feedback to understand the types of use cases for proximity control between accounts. We will use this feedback to determine how and when we might provide Availability Zone control between accounts.

Regional data transfer seems like such a small charge, why are you complicating my bill with this?

We anticipate that for most common use cases, regional data transfer will only constitute a very small portion of your monthly usage charges. There are valid use cases that involve moving large amounts of data between Availability Zones. In these cases, the regional data transfer can be a significant cost.

We try to enable as many use cases as possible while charging you only for what you use. Because of the large potential differences in the way developers could use regional data transfer, we think it is appropriate to break this cost out rather than amortize it across other charges.

If I have two instances in different Availability Zones, how will I be charged for regional data transfer?

Each instance is charged for its data in and data out. Therefore, if data is transferred between these two instances, it is charged out for the first instance and in for the second instance.

If I transfer data between Availability Zones using public IP addresses, will I be charged twice for regional data transfer (once because it crosses Availability Zones, and once because I use public IP addresses)?

No. Regional data transfer rates apply if at least one of the following cases is true, but are only charged once for a given instance even if both are true:

- The other instance is in a different Availability Zone, regardless of which type of address is used
- Public or Elastic IP addresses are used, regardless of which zone the other instance is in

Why are my Amazon EC2 resources not visible in the European region?

Amazon EC2 regions are isolated from each other. Resources such as SSH key pairs, security groups, and AMIs, are not replicated between regions. For more information, see [Resources](#) (p. 136).

Windows Instances FAQ

Can I downgrade from SQL Server Enterprise to SQL Server Enterprise 2005

Yes, Microsoft provides downgrade rights for SQL Server Enterprise.

Can I downgrade from SQL Server Enterprise to a different version of SQL Server, such as SQL Server Standard

No, Microsoft does not allow downgrades to versions of software restricted by the 90-day physical processor rule. The 90-day physical processor rule requires software to be associated with a specific processor for at least 90 days.

How can I mount or access a CD from the instance?

Select from the following:

- To create an ISO image out of the CD, upload it to your Amazon S3 bucket and download it to the instance. Then, use any standard ISO mounting tool to access it.
- To use Remote Desktop, specify the CD ROM drive letter from the **Local Resources** tab of the **Local Devices and Resources** page on the Remote Desktop client.

Monitoring, Errors, and Unexpected Behavior FAQ

How do I monitor my systems?

Amazon EC2 provides basic monitoring. You can use `DescribeInstances` to check whether an instance appears to be running. However, if you are using Amazon EC2 as your data center, you might want to set up for sophisticated monitoring on your instances, such as SNMP.

Why can't I "talk" to my instances?

There are a few common reasons for broken connectivity to your instance.

Amazon EC2 changes the state of your instance to `running` after your operating system starts booting. Depending on your AMI, there will be a delay before the instance is fully set up and functional.

If your instance has been running for several minutes, you verify you authorized the appropriate access to your host through the Amazon EC2 firewall. If you have launched your instances without specifying a security group, the `default` group is used. Permissions on the `default` group are very strict and disallow all access from the Internet and other groups. You will need to modify the permissions of

your `default` group or set up a new group with appropriate permissions. For more information, see [Network Security Concepts \(p. 18\)](#)

If this doesn't solve your issue, make sure you authorized port 22 and try to open an SSH connection with verbose output. Use the man page for the exact syntax of your system, but the command is likely to be similar to `ssh -vv root@[hostname]`. This output is very useful if you are posting to the forum.

Why did my instance terminate immediately after launch?

Launch errors can be the result of an internal error during launch or a corrupt Amazon EC2 image. Internal errors are rare, as we actively test for and isolate suspect hosts. Consult the `DescribeInstances` operation for details on why your instance failed to launch.



Note

The `ec2-describe-instances` command line tool does not provide this information. Use the `-v` flag to read the detailed SOAP response and get detailed information.

You can also attempt to launch the image again. If this proves to be a persistent problem (especially with a shared image), post to the [AWS forums](#).

I ran `shutdown` from within an `ssh` session, but my instance still shows up as running when I query it with `DescribeInstances` and I can't shell into it.

To shut down an instance, use the `TerminateInstances` call (`ec2-terminate`) on the command line. You can also use `shutdown -h`, but must verify the instance shut down using the `DescribeInstances` call.

Why are my instances stuck in a pending state (or a shutting-down state)?

This situation is rare and might be the result of a software error or misconfiguration.

We actively monitor for this; please contact us if it occurs.

Why do I get an "AuthFailure: User is not AMI creator" error when I try to register an image?

Make sure that you are using the correct user ID and certificate to create and upload the image. You must use the same ID and certificate to register the image with Amazon EC2.

Reserved Instances FAQs

What is a Reserved Instance?

Reserved Instances give you the option to make a low, one-time payment for each instance you want to reserve and in turn receive a significant discount on the hourly usage charge for that instance.

After the one-time payment for an instance, that instance is reserved for you, and you have no further obligation; you may choose to run that instance for the discounted usage rate for the duration of your term, or if and when you do not use the instance, you will not pay usage charges on it.

How is a Reserved Instance different than an On-Demand Instance?

Functionally, Reserved Instances and On-Demand instance are the same. They are launched and terminated in the same way, and they function identically once running. This makes it easy for you to seamlessly use both Reserved and On-Demand Instances without making any changes to your code. The only difference is that with a Reserved Instance, you pay a low, one-time payment and receive a lower usage rate to run the instance than with an On-Demand Instance.

How do I purchase and start up a Reserved Instance?

You purchase an EC2 Reserved Instance by calling the `PurchaseReservedInstancesOffering` API method. Launching a Reserved Instance is no different than launching an On-Demand Instance. You simply use the `RunInstances` command or launch an instance via the AWS Management Console. Amazon EC2 will optimally apply the cheapest rate that you are eligible for in the background.

How do I control which instances are billed at the Reserved Instance rate?

The `RunInstances` command does not distinguish between On-Demand and Reserved Instances. When computing your bill, our system will automatically optimize which instances are charged at the lower Reserved Instance rate to ensure you always pay the lowest amount.

How many Reserved Instances can I purchase?

You can purchase up to 20 Reserved Instances per Availability Zone each month with the EC2 APIs. If you need additional Reserved Instances, complete the [Registration Form](#).

Can a Reserved Instance that I've bought for a particular instance type (i.e. High-CPU Extra Large Instance) be applied to a different instance type that I am running (i.e. Standard Large Instance)?

No. Each Reserved Instance is associated with a specific instance type, and can only be applied to that instance type for the duration of the Reserved Instance term.

Can I move a Reserved Instance from one Region or Availability Zone to another?

No. Each Reserved Instance is associated with a specific Region and Availability Zone, which is fixed for the lifetime of the Reserved Instance and cannot be changed.

Do I need to specify an Availability Zone when I launch my instances in order to take advantage of my Reserved Instances?

Yes. When you purchase a Reserved Instance you specify the Availability Zone in which you want to reserve that instance. In order to use that Reserved Instance, you need to ensure that you launch your instance in that same Availability Zone. Additionally, you can purchase a Reserved Instance in an Availability Zone where you already have a running instance, and the Reserved Instance will automatically get applied to that existing instance.

Can I cancel a Reserved Instance?

The one-time payment for a Reserved Instances is not refundable. However, you can choose not to run or entirely stop using your Reserved Instance at any time, at which point you will not incur any further usage charges.

What happens when my Reserved Instances term comes to an end?

Any instances that you have that are still running will continue to run, but will be charged at the standard On-Demand hourly rate.

When are Reserved Instances activated?

A Reserved Instance is activated once your one-time payment has successfully been authorized. You can follow the status of your Reserved Instance on the AWS Account Activity page.

Paid AMIs FAQ



Note

You can still share AMIs without charging. Public and paid AMIs can be listed in the Resource Center.

How can I determine if a particular AMI is a paid AMI?

By describing images (ec2dim) with the "-a" flag and looking for AMIs that have a product code. For example, if you run `ec2dim -a`, the result contains an AMI with the ID `ami-bd9d78d4`. This is our Demo Paid AMI with product code `A79EC0DB`.

How can I determine if a public AMI is paid?

By describing images (ec2dim). An AMI is a paid AMI if a product code is returned. Example: run `ec2dim -a amazon`, and the AMI `ami-bd9d78d4` will be returned with a product code (`A79EC0DB`).

Is there anything that prevents a paid AMI from being rebundled? How can this be restricted?

Paid AMIs are comparable to shared AMIs with regards to rebundling and trying to restrict rebundling. If you allow a user running the AMI to see all of its contents (e.g. by giving root access to the AMI), the user could rebundle these into their own AMI.

Why can't I query a particular AMI's attributes to see if the AMI is paid?

Only the owner of an AMI can query the AMI attributes. However, anyone can tell if an AMI is paid by describing images (ec2dim). An AMI is paid if a product code is returned. Example: run `ec2dim -a amazon`, and the AMI with ID `ami-bd9d78d4` will be returned with a product code (`A79EC0DB`).

Who can use the `confirm-product-instance` command?

Only the owner of the AMI can use this command. Owners use this command with supported AMIs to determine if a supported instance with a given product code attached is up and running.

Will the product code be inherited by the rebundled AMI?

If your customer uses AWS tools to rebundle the AMI, the product code associated with the AMI is inherited by the rebundled AMI. When launching the rebundled AMI the customer is still billed for usage based on your price.



Note

This is a convenience feature and not a guarantee that the product code will always be attached to rebundled AMIs.

Note that the customer's workflow could bundle the AMI outside of Amazon EC2, or the customer could use modified versions of the AWS tools, preventing the product code from being inherited.

Will the kernel/RAM disk be inherited by the rebundled AMI?

If you rebundle an AMI, it inherits the kernel and RAM disk from the source AMI unless you specify a different kernel and RAM disk.



Note

This is a convenience feature and not a guarantee that the kernel/RAM disk will always be attached to rebundled AMIs.

I created my paid AMIs with one AWS developer account, but I want to sell them using a different AWS developer account. Can I transfer them?

No, you can't automatically transfer AMIs from one account to another. You would have to upload them again using the second AWS developer account and then register them with DevPay using that account. Alternately, you could leave the AMIs with the original account (the AMI owner account) and register them with DevPay using another AWS developer account (the product owner account). You could then use the AMI owner account to associate the product code with the AMIs. However, keep

in mind that only the product owner (and not the AMI owner in this case) can use the `ec2-confirm-product-instance` command, which confirms that an instance is running an AMI associated with the product owner's product code.

How do I prevent someone from stripping the product code from my paid AMI?

If you do not provide root access to your AMI, it cannot be rebundled. If you provide root access, our tools attempt to preserve the product code.

To increase security, we recommend that you configure your application to check the instance metadata to verify that the product code is intact.

Kernels, RAM Disks, and Block Device Mappings FAQ

What are user selectable kernels?

Amazon EC2 provides user selectable kernels which enables you to select a kernel when bundling an AMI or launching an instance. User selectable kernels are useful for keeping your instances up to date with security fixes and updates, being able to use functionality provided by new distributions, and for using specialty applications that have unique timing requirements.

How do I find user selectable kernels?

Use the `DescribeInstances` operation with the `--kernel` option. This lists all public kernels that are currently available. After locating a kernel to launch or bundle with your AMI, go to the [Resource Center](#) and search for it to determine whether there are any known issues and whether it has any dependencies.

Can I use my own kernel?

Not at present. However, as of version 2008-02-01 of the Amazon EC2 API you can use any of the kernels published by Amazon EC2 or selected vendors.

What type of dependencies do kernels have?

Kernels are most likely to require a RAM disk that contains required drivers (e.g., Xen drivers, video drivers, and so on). If you launch a kernel without a required RAM disk, it will not work properly.

How do I know a kernel/AMI combination will work together?

If you are concerned about whether the kernel/image combination will work well together, Amazon provides several AMIs that have tested combinations that you can use as a starting point for your AMIs or AMIs that you can use as a foundations for a public AMIs. If you require a certified kernel/AMI combination, you can find them as paid AMIs through organizations such as RedHat. For more information, see [Paying for AMIs \(p. 78\)](#).

Error Messages FAQ

Why do I get an "InsufficientInstanceCapacity" error when I try to launch an instance?

This error indicates that we do not currently have enough available capacity to service your request.

If you are requesting a large number of instances, there might not be enough server capacity to host them. You can try again later or specify a smaller number of instances.

Why do I get an "InstanceLimitExceeded" error when I try to launch an instance?

This error indicates you reached your concurrent running instance limit. For new users during the public beta, the limit is 20.

If you need additional capacity, please contact us at aws@amazon.com.

Why can't I retrieve my instance-specific data from within a running instance when querying <http://169.254.169.254/2009-04-04/>?

The Parameterized Launches feature is available to instances that were launched after the feature was released. If you launched your instance before this, the data will not be available. If you want to use this functionality, relaunch your instances.

If you still experience problems retrieving the data after relaunching your instance, check the following:

- Verify you are using the correct base URI (<http://169.254.169.254/2009-04-04/>)
- Verify you are using the correct URI for the data you are trying to retrieve. Depending on the data, a trailing '/' might be required
- Verify you specified launch data when launching your instances. If not, you will get a HTTP error response (404) when trying to retrieve the user data



Note

Instance metadata is always available, even if you do not specify it at instance launch.

Why do I get keep getting "Request has expired" errors?

To reduce the risk of replay attacks, our requests include a timestamp. This and the most important parts of the request are signed to ensure the message (including the timestamp) cannot be modified without detection.

If the difference between the timestamp in the request and the time on our servers is larger than 5 minutes, the request is too old (or too new) and an error is returned.

You need to ensure that your system clock is accurate and configured to use the correct time zone. For more information, go to [NTP](#).

Miscellaneous FAQ

What runlevel do instances start in?

All Linux instances are started in runlevel 4, regardless of the instance configuration.

Can I perform root file system booting from an EBS volume?

At this time, there is no way to directly boot off an EBS volume. However, check the forums for information on how to create a bootstrap AMI that runs an instance and changes the root file system to an Amazon EBS volume.

Are there any special requirements to use FTP?

The File Transfer Protocol (FTP) has a PORT command by which a client sends its address back to the server. The server then connects to the client at that address to send the file data. If the client looks up its own internal address and sends this to the server, the connection will fail. In this specific case, there are two solutions to the problem. First, configure the client to send its public IP address. Second, the client can use "passive FTP" which makes connections only to the server, rather than from the server to the client. In general, applications which encode local addresses and port numbers in data sent to external servers might have problems with NAT. Care must always be taken to send the public address, rather than the internal one.

We recommend using passive mode unless it is not supported by the FTP server.

How can I verify the authenticity of the Amazon EC2 client tools?

Our published client tools are signed using GPG (<http://www.gnupg.org>), an implementation of the OpenPGP security standard. This allows you to verify the integrity of the packages we publish.

Before you can begin verification, you need to set up a GPG identity. For more information, go to <http://www.gnupg.org>.

To quickly get set up

1. Enter the following:
`# gpg --gen-key`
2. You must download the public portion of our signing key, and import it into your GPG and RPM keychains. To get the signing key, visit <https://aws.amazon.com/ec2/public-key.asc>.
3. Import the public signing key into GPG:
`# gpg --import public-key.asc`
4. Import the public signing key into RPM:
`# rpm --import public-key.asc`
Your system is now set up to verify our signatures.
5. To verify ZIP archives, download an archive and its signature file and run the following:
`# gpg --verify signature_file signed_archive`
6. To verify RPM Packages, download an RPM package and run the following:
`# rpm --checksig rpm_file`

Appendix

Topics

- [Resources](#) (p. 136)
- [Metadata Categories](#) (p. 137)
- [Windows Configuration Service](#) (p. 138)

Resources

The following table describes which Amazon EC2 resources are global, regional, or Availability Zone-based.

Resource	Type	Description
AWS Account	Global	You use the same AWS account in all regions.
DevPay Product Codes	Global	You use the same DevPay product codes throughout all regions.
Amazon EC2 System Identifiers	Regional	Includes the AMI ID, Instance ID, EBS Volume ID, EBS Snapshot ID, and so on.
Instances	Availability Zone	Instances are tied to Availability Zones. However, the instance ID is tied to the region.
AMIs	Regional	AMIs are tied to the region where its files are located within Amazon S3.
Security Groups	Regional	Security groups are not copied across regions. Instances within the region cannot communicate with instances outside the region using group-based firewall rules. Traffic from instances in another region is seen as WAN bandwidth.
SSH Key Pairs	Regional	Key pairs (to connect to instances) are region-specific.
User-Supplied Identifiers	Regional	Includes security group names, SSH key pair names, and so on. Although you can create the same names in multiple regions, they have no relationship to each other.

Resource	Type	Description
Elastic IP Addresses	Regional	Elastic IP addresses are tied to a region and cannot be mapped across regions.
EBS Volumes	Availability Zone	An Amazon EBS volume must be located within the same Availability Zone as the instance to which it attaches.
EBS Snapshots	Regional	Snapshots are tied to regions and can only be used for volumes within the same region.

Metadata Categories

The data available to instances is categorized into metadata and user-supplied data.

Metadata is specific to an instance and is described in the following table.

Data	Description	Version Introduced
ami-id	The AMI ID used to launch the instance.	1.0
ami-launch-index	The index of this instance in the reservation (per AMI).	1.0
ami-manifest-path	The manifest path of the AMI with which the instance was launched.	1.0
ancestor-ami-ids	The AMI IDs of any instances that were rebundled to create this AMI.	2007-10-10
block-device-mapping	Defines native device names to use when exposing virtual devices.	2007-10-10
instance-id	The ID of this instance.	1.0
instance-type	The type of instance to launch. For more information, see Instance Types (p. 12).	2007-08-29
local-hostname	The local hostname of the instance.	2007-01-19
local-ipv4	Public IP address if launched with direct addressing; private IP address if launched with public addressing.	1.0
kernel-id	The ID of the kernel launched with this instance, if applicable.	2008-02-01
placement/availability-zone	The Availability Zone in which the instance launched.	2008-02-01
product-codes	Product codes associated with this instance.	2007-03-01
public-hostname	The public hostname of the instance.	2007-01-19
public-ipv4	The public IP address	2007-01-19
public-keys/	Public keys. Only available if supplied at instance launch time	1.0

Data	Description	Version Introduced
ramdisk-id	The ID of the RAM disk launched with this instance, if applicable.	2008-02-01
reservation-id	ID of the reservation.	1.0
security-groups	Names of the security groups the instance is launched in. Only available if supplied at instance launch time	1.0

User-supplied data is treated as opaque data: what you give us is what you get back.



Note

- All instances launched together get the same user-supplied data. You can use the AMI launch index as an index into the data.
- User data is limited to 16K. This limit applies to the data in raw form, not base64 encoded form.
- The user data must be base64 encoded before being submitted to the API. The API command line tools perform the base64 encoding for you. The data is in base64 and is decoded before presented to the instance.

Windows Configuration Service

Before bundling an instance, you can configure the instance using the EC2Config service. The EC2Config service sets up and initializes the instance during startup, prepares the service for bundling, and manages the event log.

There are three EC2Config files that you can modify: `Config.xml`, `BundleConfig.xml`, and `EventLogConfig.xml`.



Note

By default, the EC2Config service is installed on all Amazon EC2 public Windows AMIs (Program Files\Amazon\Ec2ConfigSetup\).

Config.xml File

This section describes the `Config.xml` file.

Config.xml File

- **Ec2SetPassword**—Generates a new password on instance launch. By default, Amazon EC2 disables this after the first launch. To continue generating random passwords, set this to `Enabled`.
- **Ec2SetComputerName**—When enabled, sets the hostname to the internal DNS name of the instance and reboots.
- **Ec2InitializeDrives**—Initializes and formats the instance stores during startup. For more information on instance storage, see [Instance Storage \(p. 13\)](#).
- **Ec2ConfigureRDP**—Sets up a self-signed certificate on the instance, so users can securely access the instance using Remote Desktop.

- **Ec2OutputRDPcert**—Copies the Remote Desktop certificate information to the console, so the user can verify it against the thumbprint
- **Ec2EventLog**—Puts eventlog entries on the console based on the configuration of the eventlogconfig file.

BundleConfig.xml File

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` plugin, and shutting down the instance for bundling. To not use sysprep, change the value of `SetSysprep` to `No`. To not set the Remote Desktop Certificate, set the value of `SetRDPCertificate` to `No`.

EventLogConfig.xml File

This section describes the `EventLogConfig.xml` file.

EventLogConfig.xml File

- **Category**—Event log key to monitor.
For more information, go to the [Microsoft Web Site](#),
- **ErrorType**—The type of error (i.e., Error, Warning, Information).
For more information, go to the [Microsoft Web Site](#),
- **AppName**—The event source or application that logged the event.
For more information, go to the [Microsoft Web Site](#),
- **NumEntries**—The number of events stored for this category.
- **LastMessageTime**—To prevent the same message from being pushed repeatedly, the service updates this every time it pushes a message.

Example

The following are examples of event log entries. The first entry pushes the last 3 errors from system category, regardless of the application that generated the `LastMessage` entry. The second entry pushes the last 3 error entries written by `Ec2Config` generated after `LastMessageTime`.

```
<EventLogConfig>
  <Event>
    <Category>System</Category>
    <ErrorType>Error</ErrorType>
    <NumEntries>3</NumEntries>
    <LastMessageTime>2008-09-10T00:00:00.000Z</LastMessageTime>
    <AppName></AppName>
  </Event>
  <Event>
    <Category>Application</Category>
    <ErrorType>Error</ErrorType>
    <NumEntries>3</NumEntries>
    <LastMessageTime>2008-09-10T00:00:00.000Z</LastMessageTime>
    <AppName>Ec2Config</AppName>
  </Event>
</EventLogConfig>
```

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.
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 user.
group	See security group .

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 while others are designed for CPU-intensive applications.
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. So yes, Amazon has bigger bytes.
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	The disk storage associated with an instance. In the event an instance fails or is terminated (not simply rebooted), all content on the instance store is deleted.
group	Also known as a security group, groups define firewall rules that can be shared among a group of instances that have similar security requirements. The group is specified at instance launch.
launch permission	AMI attribute allowing users to launch an AMI
Linux	Amazon EC2 instances are available for many operating platforms, including Linux, Solaris, Windows, and others.
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 users 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 all AWS services, users pay only for the compute and storage they use for their own 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.
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.
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.
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. So yes, Amazon has bigger 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 Conventions

This section lists the common typographical and symbol use conventions for AWS technical publications.

Typographical Conventions

This section describes common typographical use conventions.

Convention	Description/Example
Call-outs	A call-out is a number in the body text to give you a visual reference. The reference point is for further discussion elsewhere. You can use this resource regularly. 1
Code in text	Inline code samples (including XML) and commands are identified with a special font. You can use the command <code>java -version</code> .
Code blocks	Blocks of sample code are set apart from the body and marked accordingly. <pre># ls -l /var/www/html/index.html -rw-rw-r-- 1 root root 1872 Jun 21 09:33 /var/www/html/ index.html # date Wed Jun 21 09:33:42 EDT 2006</pre>
Emphasis	Unusual or important words and phrases are marked with a special font. You <i>must</i> sign up for an account before you can use the service.
Internal cross references	References to a section in the same document are marked. See Document Conventions (p. 143) .
Logical values, constants, and regular expressions, abstracta	A special font is used for expressions that are important to identify, but are not code. If the value is <code>null</code> , the returned response will be <code>false</code> .

Amazon Elastic Compute Cloud User Guide Typographical Conventions

Convention	Description/Example
Product and feature names	Named AWS products and features are identified on first use. Create an <i>Amazon Machine Image</i> (AMI).
Operations	In-text references to operations. Use the <code>GetHITResponse</code> operation.
Parameters	In-text references to parameters. The operation accepts the parameter <i>AccountID</i> .
Response elements	In-text references to responses. A container for one <code>CollectionParent</code> and one or more <code>CollectionItems</code> .
Technical publication references	References to other AWS publications. If the reference is hyperlinked, it is also underscored. For detailed conceptual information, see the <i>Amazon Mechanical Turk Developer Guide</i> .
User entered values	A special font marks text that the user types. At the password prompt, type MyPassword .
User interface controls and labels	Denotes named items on the UI for easy identification. On the File menu, click Properties .
Variables	When you see this style, you must change the value of the content when you copy the text of a sample to a command line. <code>% ec2-register <your-s3-bucket>/image.manifest</code> See also Symbol Conventions (p. 145) .

Symbol Conventions

This section describes the common use of symbols.

Convention	Symbol	Description/Example
Mutually exclusive parameters	(Parentheses and vertical bars)	Within a code description, bar separators denote options from which one must be chosen. <code>% data = hdfread (start stride edge)</code>
Optional parameters XML variable text	[square brackets]	Within a code description, square brackets denote completely optional commands or parameters. <code>% sed [-n, -quiet]</code> Use square brackets in XML examples to differentiate them from tags. <code><CustomerId>[ID]</CustomerId></code>
Variables	<arrow brackets>	Within a code sample, arrow brackets denote a variable that must be replaced with a valid value. <code>% ec2-register <your-s3-bucket>/image.manifest</code>

Index

A

- accessing instances, 83
- addressing, 17, 87
- Amazon CloudWatch
 - concepts, 24
 - using, 115
- Amazon DevPay, 56
- Amazon EBS
 - concepts, 21
- AMIs
 - bundling, 44
 - creating, 28
 - paid, 78
 - shared, 76
 - finding, 77
 - security, 77
 - sharing, 49
- appendix, 136
- audience, 1
- Auto Scaling
 - concepts, 24
 - using, 115
- Availability Zones, 19, 102, 127

B

- batch processing, 6
- best practices, 11
- block device mapping, 133
- bundling AMIs, 44

C

- categories, 137
- Census data, 25
- changes to Amazon EC2, 4
- CloudWatch
 - concepts, 24
 - using, 115
- computation building block, 11
- compute resources, measuring, 15
- concepts, 10
 - Amazon EBS, 21
 - block storage, 21
 - public data sets, 25
 - Reserved Instances, 14
 - Windows, 14
- console output, 81
- CPU, 16
- creating AMIs, 28
- creating paid AMIs, 56

D

- data retrieval, 71
- data sets, 25, 117

- device mapping, 133
- DevPay, 56
- disk
 - performance, 76
 - RAID, 76
- DNS, internal, 17

E

- elastic IP addresses, 17, 87
- Elastic Load Balancing
 - concepts, 24
 - using, 115
- errors, 129
 - messages, 133

F

- FAQs, 121
 - Availability Zones, 127
 - block device mapping, 133
 - errors, 129
 - general, 121
 - instance types, 123
 - IP addresses, 125
 - kernels, 133
 - miscellaneous, 134
 - monitoring, 129
 - operations, 122
 - paid AMIs, 131
 - proximity, 127
 - RAM disk, 133
 - Reserved Instances, 130
 - unexpected behaviors, 129
 - Windows, 129
- fault tolerance, 20
 - using, 115

G

- general information, 121
- glossary, 140

H

- Human Genome Project data, 25

I

- I/O resources, 16
- instance store, 13, 75
- instance types, 123
- instances
 - accessing, 83
 - addressing, 17, 87
 - launching, 64
 - metadata, 71
 - rebooting, 81
 - security, 17, 87
 - sizes, 12
 - storage, 13, 75

- types, 12
- usage, 11
- introduction, 6
- IP address information, 125

K

- kernels, 133

L

- launch data, security, 71
- launch index, example, 73
- leases, 14, 119
- load balancing, 20
 - concepts, 24
 - using, 115
- locality, 19, 102

M

- mapping, block device, 133
- memory, 16
- metadata, 71
 - categories, 137
 - retrieval, 71
- miscellaneous FAQs, 134
- monitoring, 20
 - using, 115
- monitoring information, 129

N

- NAT, 17, 87
- network security, 18, 94
- new features, 4

O

- operations
 - information, 122
- output, console, 81
- overview, 6

P

- paid AMIs
 - creating, 56
 - information, 131
- Paid AMIs, 78
- performance, optimization, 76
- permissions, 11
- private addresses, 17, 87
- proximity, 19, 102, 127
- public addresses, 17, 87
- public data sets
 - concepts, 25
 - using, 117

R

- RAID, 76

- RAM disk, 133
- reboot, 81
- remote access, 83
- Remote Desktop, 83
- required knowledge, 1
- Reserved Instances
 - concepts, 14
 - information, 130
 - using, 119
- resources, 2
 - I/O, 16
 - measuring, 15
- retrieving metadata, 71
- retrieving user data, 73

S

- scalability, 20
 - using, 115
- scalable applications, 6
- security, 18, 94
- service overview, 6
- shared AMIs, 76
 - finding, 77
 - security, 77
- sharing AMIs, 49
- sizes of instances, 12
- SQL, 129
- SSH, 83
- static IPs, 17, 87
- storage, 13, 16, 75
 - locations, 13
- suggestions, 11

T

- temporary events, 6
- types of instances, 12

U

- unexpected behavior information, 129
- US Census data, 25
- user data, retrieval, 73
- using, 27
 - public data sets, 117
 - Reserved Instances, 119

W

- Wikipedia data, 25
- Windows, 129
 - concepts, 14
- Windows AMIs, 59

Z

- zones, availability, 127