Welcome

Topics

- Who Should Read This Guide
- Reader Feedback
- How This Guide Is Organized
- Amazon EC2 Resources

This is the *Amazon Elastic Compute Cloud* Developer 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 picks up where the <u>Amazon Elastic Compute Cloud Getting Started Guide</u> ends and provides you with the information to create more sophisticated Amazon Machine Images (AMIs), using advanced service features.

Most examples in this guide use the SOAP and Query APIs. If you are interested in using the AWS Management Console or the command line tools, refer to the <u>Amazon Elastic Compute Cloud User Guide</u>.

Required Knowledge and Skills

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

- XML (For an overview, go to the <u>W3 Schools XML Tutorial</u>)
- Basic understanding of web services (go to <u>W3 Schools Web Services</u> <u>Tutorial</u>)
- Query or SOAP
- One or more programming languages

You should also have worked through the <u>Amazon Elastic Compute Cloud Getting Started Guide</u>, 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
Amazon EC2 concepts and an overview of major Amazon EC2 features.	Amazon EC2 Concepts
Describes 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
Answers to commonly asked questions.	Technical FAQ
Amazon EC2 terms.	Glossary
Typographic and symbol conventions.	Document Conventions

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.

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 Technical FAQ	The FAQ covers the top questions developers have asked about this product.
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.
AWS Management Console	The console allows you to perform most of the functions of Amazon EC2 and other AWS products without programming.
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.

Amazon EC2 Product Information	The primary web page for information about Amazon EC2.
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 2009-10-31 release of Amazon EC2. This guide was last updated on December 09, 2009.

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

Change	Description	Release Date
Windows 2008	Amazon EC2 now supports Windows 2008. For more information, go to the Amazon Elastic Compute Cloud Detail Page.	9 December 2009
AMIs Backed by Amazon EBS	Amazon EC2 AMIs can now use Amazon EBS volumes as their root devices. These are easier to create than AMIs that use instance stores as their root devices, launch faster, support sizes up to 1 TiB, and use persistent storage.	2 December 2009
Amazon EC2 in Northern California	Amazon EC2 is now available in the Northern California. For more information, go to the <u>Amazon Elastic Compute Cloud Detail Page</u> .	2 December 2009

Introduction to Amazon Elastic Compute Cloud

Topics

•

What Is Amazon EC2?

- Service Highlights
- <u>Features</u>
- Popular Uses for Amazon EC2
- Amazon EC2 Charges

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.

Service Highlights

- **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. You can stop your instance while retaining the data on your boot partition and subsequently restart the same instance using web service APIs. You also have access to console output of your instances.
- **Flexible**—You have the choice of multiple instance types, operating systems, and software packages. Amazon EC2 allows you to select a configuration of memory, CPU, and instance storage, and the boot partition size that is optimal for your choice of operating system and application. For example, your choice of operating systems includes numerous Linux distributions, Microsoft Windows Server and OpenSolaris.
- 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 predictably commissioned. The service runs within Amazon's proven network infrastructure and data centers. The Amazon EC2 Service Level Agreement commitment is 99.95% availability for each Amazon EC2 Region.
- **Secure**—Amazon EC2 provides numerous mechanisms for securing your compute resources.
 - Amazon EC2 includes web service interfaces to configure firewall settings that control network access to and between groups of

instances.

- When launching Amazon EC2 resources within Amazon Virtual Private Cloud (Amazon VPC), you can isolate your compute instances by specifying the IP range you wish to use, and connect to your existing IT infrastructure using industry-standard encrypted IPsec VPN.
- **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

Amazon EC2 provides a number of powerful features for building scalable, failure resilient, enterprise class applications, including:

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 leveraged as an Amazon EC2 instance's boot partition or attached to a running Amazon EC2 instance and are exposed as a standard block devices. When used as a boot partition, Amazon EC2 instances can be stopped and subsequently restarted, enabling you to only pay for the storage resources used while maintaining your instance's state. 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. You can also easily share these snapshots with co-workers and other AWS developers.
- 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. The Amazon EC2 Service Level Agreement commitment is 99.95% availability for each Amazon EC2 Region. Amazon EC2 is currently available in two three Regions: the US-East (Northern Virginia) Region and the US-West

(Northern California) Region in the United States, and the EU (Ireland) Region in Europe.

- 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.
- Amazon Virtual Private Cloud

 —Amazon VPC is a secure and seamless bridge between a company's existing IT infrastructure and the AWS cloud. Amazon VPC enables enterprises to connect their existing infrastructure to a set of isolated AWS compute resources via a Virtual Private Network (VPN) connection, and to extend their existing management capabilities such as security services, firewalls, and intrusion detection systems to include their AWS resources.
- Amazon CloudWatch—Amazon CloudWatch is a web service that provides monitoring for AWS cloud resources, starting with Amazon EC2. It provides you with visibility into resource utilization, operational performance, and overall demand patterns—including metrics such as CPU utilization, disk reads and writes, and network traffic. To use Amazon CloudWatch, simply select the Amazon EC2 instances that you'd like to monitor; within minutes, Amazon CloudWatch will begin aggregating and storing monitoring data that can be accessed using web service APIs or Command Line Tools.
- **Auto Scaling**—Auto Scaling allows you to automatically scale your Amazon EC2 capacity up or down according to conditions you define. With Auto Scaling, you can ensure that the number of Amazon EC2 instances you're using scales up seamlessly during demand spikes to maintain performance, and scales down automatically during demand lulls to minimize costs. Auto Scaling is particularly well suited for applications that

experience hourly, daily, or weekly variability in usage. Auto Scaling is enabled by Amazon CloudWatch and available at no additional charge beyond Amazon CloudWatch fees.

• Elastic Load Balancing—Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve even greater fault tolerance in your applications, seamlessly providing the amount of load balancing capacity needed in response to incoming application traffic. Elastic Load Balancing detects unhealthy instances within a pool and automatically reroutes traffic to healthy instances until the unhealthy instances have been restored. You can enable Elastic Load Balancing within a single Availability Zone or across multiple zones for even more consistent application performance. Amazon CloudWatch can be used to capture a specific Elastic Load Balancer's operational metrics, such as request count and request latency, at no additional cost beyond Elastic Load Balancing fees.

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.

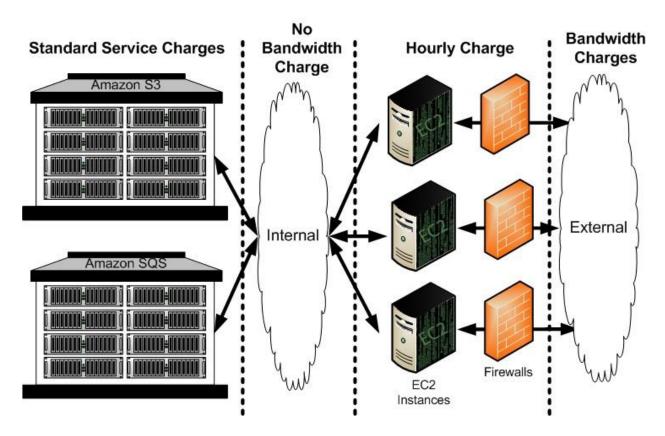
This includes video and image processing, financial data processing, and science and research applications.

• **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 <u>Amazon EC2</u> <u>Product Page</u>.

Amazon EC2 Concepts

Topics

•

AMI and Instance Concepts

- Amazon EC2 Flow
- Amazon EC2 Credential Concepts
- Instance Addressing Concepts
- Network Security Concepts
- Region and Availability Zone Concepts
- Failure Resilient Application Concepts
- Amazon Virtual Private Cloud
- Public Data Set Concepts

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.

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 all the software to act as a web server (e.g., Linux, Apache, and your web site) or it might contain all the software to act as a Hadoop node (e.g., Linux, Hadoop, and a custom application).

You launch one or more instances of an AMI. An instance might be one web server within a web server cluster or one Hadoop node.

AMIs

An Amazon EC2 instance can be launched from an AMI backed by Amazon S3 or an AMI backed by Amazon EBS. Instances launched from AMIs backed by Amazon S3 use an instance store as the root device (e.g., / or C:\). Instances launched from AMIs backed by Amazon EBS use Amazon EBS volumes as their root devices.

The following table describes the differences between AMIs backed by Amazon S3 and AMIs backed by Amazon EBS.

Root Device	Instance Store	Amazon EBS
Boot Time	Usually Less than 5 minutes	Usually Less than 1 minute
Size Limit	10 GiB	1 TiB
Location	Instance storage	Amazon EBS volume
Data Persistence	Data persists for the life of the instance; non-root devices can use Amazon EBS	Data persists on instance failure and can persist on instance termination
Upgrading	Instance attributes are fixed for the life of an instance	The kernel, instance type, and ramdisk can be changed while the instance is stopped.
Charges	Instance usage and AMI storage	Instance usage, Amazon EBS volume usage, and AMI Storage
AMI Creation	Requires installation and use of AMI tools	Uses a single command/call
Stopped State	Cannot be in stopped state; instances are running or not	Can be placed in stopped state where instance is not running, but state is maintained in Amazon EBS

Public AMIs are available from 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.

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 and How to Share AMIs.

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

Preparing and Creating an AMI

To use a file system image with Amazon EC2, you must prepare and create it as an AMI.

An Amazon EC2 instances can use its instance store (Amazon S3-backed) as its root device or an Amazon EBS volume (Amazon EBS-backed).

The creation process for an AMI that uses an instance store as its root device 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



After completion, you must register the AMI.

The creation process for an AMI that uses an Amazon EBS volume as its root device uses one of the following methods:

• Creates snapshots for all attached Amazon EBS volumes and creates an AMI using a single command/call

• Creates an AMI from a snapshot

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 <u>Amazon EC2 Instance Request Form</u> 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. However, if an instance fails that uses an Amazon EBS volume as its root device, the data remains available.



Note

Instances that use Amazon EBS volumes as their root devices can delete their volumes on termination or leave them running. If an instance fails that uses an Amazon EBS volume as its root device, the volume continues running and can be mounted by another instance.

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**.

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 or use Amazon EBS.
- 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.

For more information, go to <u>Amazon CloudWatch Product Page</u>.

• 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 three families: standard, High-CPU, and High-Memory. 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. High-Memory instances have proportionally more memory resources and are well suited for high throughput applications, such as database and memory caching 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.



Tip

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. A 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.

Туре	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
High- Memory Double Extra Large	13 EC2 Compute Units (4 virtual cores with 3.25 EC2 Compute Units each)	34.2 GB	850 GB instance storage (1 x 840 GB plus 10 GB root partition)	64-bit	High	m2.2xlarge
High- Memory Quadruple Extra Large	26 EC2 Compute Units (8 virtual cores with 3.25 EC2 Compute Units each)	68.4 GB	1690 GB instance storage (2 x 840 GB plus 10 GB root partition)	64-bit	High	m2.4xlarge



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.



We strongly recommend using the 2.6.18 Xen stock kernel with High-CPU and High-Memory 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.

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, the instance is terminated, or the instance is stopped, the data is lost.

If you need a permanent storage solution, we recommend using Amazon EBS. For more information, see <u>Amazon Elastic Block Store</u>.



Note

If you do not use Amazon EBS, make sure to back 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	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)	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.



Note

By default, instance stores are not exposed to Amazon EBS-backed instances. However, they are still available.

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.

Reserved Instance Process

- 1 Choose a Region where you want to run the instance.
- 2 Search for offerings. To limit the results returned, you can specify the instance type or Availability Zone.
- 3 Purchase offerings that meet your requirements.
- 4 Run instances of the purchased instance type in the correct Region and Availability Zone.



Note

For information about pricing, refer to the **Amazon EC2 Product Page**.

For information on using Reserved Instances, see <u>Reserving Amazon EC2</u> Instances.

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.
- **Bundling**—Windows instances use different bundling procedures. For more information, see Preparing and Creating AMIs.

Amazon EC2 Running Windows

As part of this service, Amazon EC2 instances can now run Microsoft Windows Server 2003 or Microsoft Windows Server 2008. The Windows AMIs provide you with all standard Microsoft Windows Server functionality.



To get started using Windows instances, we recommend using the <u>AWS</u> <u>Management Console</u>.

Windows AMI

Amazon EC2 currently provides the following Windows AMIs:

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

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 <u>Using Instance Addressing</u>.
- 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.



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 <u>Instance page</u>.

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.

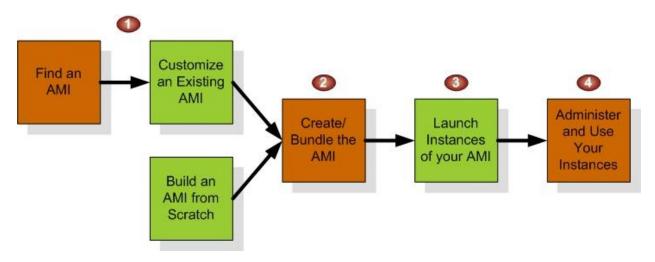
The different instance types provide higher or lower minimum performance from the shared resources depending on their size. Each of the instance types has an I/O performance indicator (moderate or high). Instance types with high I/O performance have a larger allocation of shared resources. Allocating 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

- Preparing and Creating AMIs
- Launching and Using Instances
- Accessing Instances

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 Preparing an AMI.
- 2 For an AMI that uses an instance store as its root devices, you follow the bundling process and register it. For an AMI that uses an Amazon EBS volume as its root device, you simply use the create image command on a running instance. Amazon EC2 returns an AMI ID, so you can launch as many instances of the AMI as you want. See <u>Creating an AMI</u>.
- 3 You launch one or more instances of your AMI. See <u>Launching and Using Instances</u>.
- 4 You administer and use your instances as you would with any servers.

Related Topics

- Preparing and Creating AMIs
- Launching and Using Instances
- Accessing Instances

Amazon EC2 Credential Concepts

This section describes major tasks and the credentials required to perform them.

Access Your Account

To sign up for services, view your bills, perform account-based tasks, and get many of your security credentials, you will need your standard Amazon login and password. For more information, see How to Log In with Your Amazon Login and Password.



We also provide multi-factor authentication, which requires a physical device and passcode to login to your AWS account. For more information, go to http://aws.amazon.com/mfa.

Credentials to Launch and Administer Amazon EC2 Instances

To use Amazon EC2, you must get the right set of credentials. The set of credentials you need depends on how you access Amazon EC2. For information about getting credentials, see <u>Using Credentials</u>.

AWS Management Console

To launch and administer Amazon EC2 instances through the AWS Management Console, you only need the Amazon login and password. For more information, see <u>How to Log In with Your Amazon Login and Password</u>.

Query and Third Party UI Tools

Launching and administering Amazon EC2 instances through the Query APIs and many UI-based tools (e.g., ElasticFox) requires the Access Key ID and Secret Access Key. For more information, see How to Get Your Access Key ID and Secret Access Key.

SOAP and EC2 Command Line Tools

Launching and administering Amazon EC2 instances through the SOAP APIs requires an X.509 certificate and private key, which can be generated by AWS. For more information, see How to Create an X.509 Certificate and Private Key.

Credentials to Connect to Amazon EC2 Instances

To connect to your instances, you need the following:

Credentials List

- **Amazon EC2 Key Pair**—Enables you to connect to instances through SSH. For more information, see <u>SSH Key Pair</u>.
- **Windows administrator password (Windows only)**—provides the "first-use" password which enables you to connect to a Windows instance through Remote Desktop. For more information, see Windows Administrator Password.

Credentials Needed for Sharing with Others

To enable other AWS accounts to use your Amazon EC2 AMIs and Amazon EBS snapshots, you need their AWS Account IDs. For information on how to get the AWS Account ID associated with your account,see ???.

Credentials Needed for Bundling Instances

To bundle Linux/UNIX instances, you need the following:

Credentials List

- Access Key ID and Secret Access Key—Enables you to connect to instances through SSH. For more information, see SSH Key Pair.
- **X.509 certificate and private key**—provides the "first-use" password which enables you to connect to a Windows instance through Remote Desktop. For more information, see Windows Administrator Password.
- **AWS Account ID**—provides the "first-use" password which enables you to connect to a Windows instance through Remote Desktop. For more information, see Windows Administrator Password.

To bundle Windows instances, you need your Amazon login and password to access the AWS Management Console. For information about the Amazon login and password, see How to Log In with Your Amazon Login and Password.



For information about bundling instances, see ???.

- <u>Using Credentials</u>
- <u>???</u>
- Launching and Using Instances
- Accessing Instances

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.



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 <u>Elastic IP Addresses</u>.

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 <u>RFC</u> 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.

- Elastic IP Addresses
- <u>Using Instance Addressing</u>

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.



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.



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).

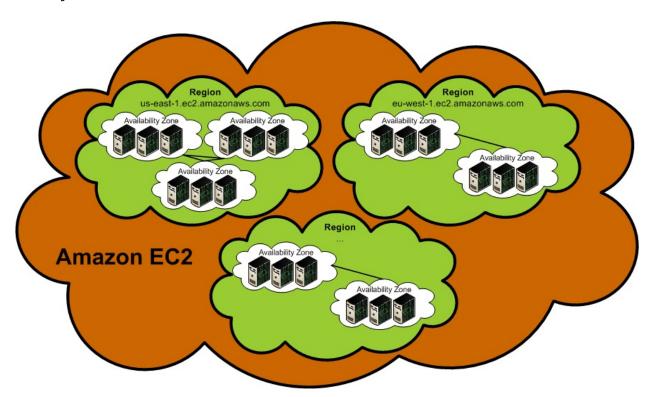
• <u>Using Network Security</u>

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 Region URL endpoint. For example, to access the US-East Region (default), you make service calls to the us-east-1.ec2.amazonaws.com service endpoint. To access the EU Region, you make service calls to the eu-west-1.ec2.amazonaws.com service endpoint.

Amazon EC2 supports the following endpoints:

- **us-east-1.ec2.amazonaws.com**—Endpoint located in the US-East (Northern Virginia) Region
- **us-west-1.ec2.amazonaws.com**—Endpoint located in the US-West (Northern California) Region
- **eu-west-1.ec2.amazonaws.com** —Endpoint located in the EU (Ireland) Region

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.



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.

- <u>Using Regions and Availability Zones</u>
- Region and Availability Zone FAQ

Failure Resilient Application Concepts

•

Using Amazon Elastic Block Store

- Auto Scaling
- Elastic Load Balancing
- Amazon CloudWatch

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.

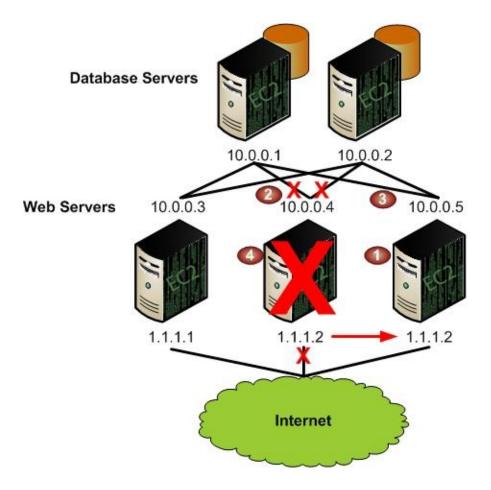
To solve this problem, Amazon EC2 provides elastic IP addresses. Elastic IP addresses are static IP addresses designed for dynamic cloud computing. Elastic IP addresses are associated with your account, not specific instances. Any elastic IP addresses that you associate with your account remain associated 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.



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).



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. Additionally, you can share snapshots with specific users or make a snapshot public.

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

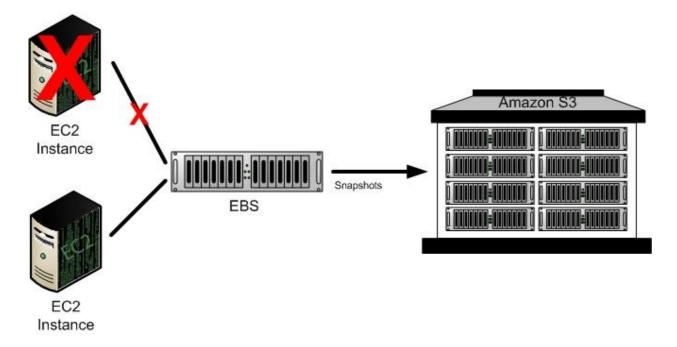
 A method for instantiating new volumes that contain the exact contents of a snapshot

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.

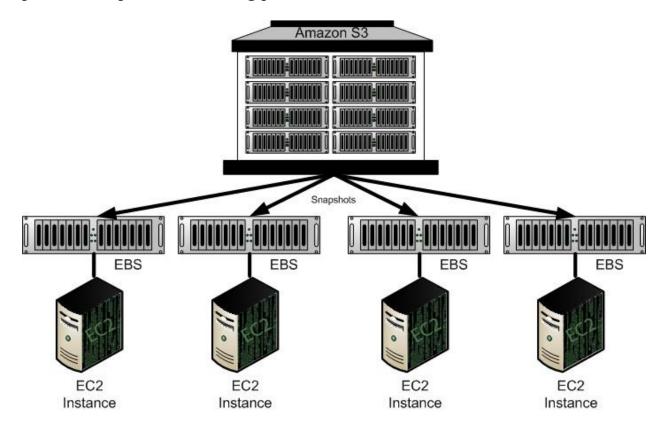


Launch Process

- 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.
- 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.
- 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

Root Partition

You can use an Amazon EBS volume as the root device on Linux/UNIX and Windows instances. This enables you to have a root device up to 1 TiB in size.

Additionally, you can resize the root device at any time and, in the event of instance failure, mount the device from another instance.

Launch Process

- 1 You find an existing AMI backed by Amazon EBS, make changes, and create a new AMI.
- 2 If the root device becomes too small, you stop the instance and restart it using a larger Amazon EBS volume.
- 3 If the instance fails, you launch a new instance and mount the root device of the failed instance.

Large Data Sets

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

• <u>Using Amazon Elastic Block Store</u>

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*.

Amazon Virtual Private Cloud

Amazon Virtual Private Cloud is a secure and seamless bridge between a company's existing IT infrastructure and the AWS cloud. Amazon VPC enables enterprises to connect their existing infrastructure to a set of isolated AWS compute resources via a Virtual Private Network (VPN) connection, and to extend their existing management capabilities such as security services, firewalls, and intrusion detection systems to include their AWS resources.

With Amazon Virtual Private Cloud, you create a VPC by first defining its IP address space. The IP addresses in this address space are private and form a network that is isolated at a packet-routing level from any other network, including the Internet.

You then create subnets, which are segments of a VPC's IP address space. These let you separate the Amazon EC2 instances in the VPC based on security and operational requirements. If you create more than one subnet in a VPC, they're attached to each other by a logical router, in a star topology.

To connect to a VPC, you create a VPN connection, which is a VPN tunnel between a VPC and a data center, home network, or co-location facility. You configure your existing network to route all VPC-bound traffic to the gateway that anchors your end of the VPN connection.

With a VPN connection established, you can launch Amazon EC2 instances into a VPC's subnets; with the appropriate security policy, these instances now appear on your existing network.

VPC traffic bound for the Internet is routed over the VPN to your existing network, where it can be examined by pre-existing network security services, such as firewalls and intrusion detection systems, before exiting your existing network perimeter to the Internet. This is particularly valuable if you're using specialized network appliances and software to enforce security policies.

Amazon VPC has its own set of documentation to describe how to create and use your VPC, subnets, and VPN connection. The following table gives links to the Amazon VPC guides.

Description	Documentation
How to get started using Amazon VPC	Amazon Virtual Private Cloud Getting Started Guide
How to use Amazon VPC through the SOAP and Query APIs	Amazon Virtual Private Cloud Developer Guide
Complete descriptions of all the Amazon VPC commands	Amazon Virtual Private Cloud Command Line Reference
Complete descriptions of the Amazon VPC API operations, data types, and errors	Amazon Virtual Private Cloud API Reference
Information for the network administrator who needs to configure the gateway at your end of the VPN connection	Amazon Virtual Private Cloud Network Administrator Guide

• <u>Using Amazon Virtual Private Cloud</u>

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.



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

• <u>Using Public Data Sets</u>

Using Amazon EC2

Topics

•

Using Credentials

- Preparing and Creating AMIs
- Launching and Using Instances
- Accessing Instances
- Using Instance Addressing
- Using Network Security
- Using Regions and Availability Zones
- Using Amazon Elastic Block Store
- Using Auto Scaling, Elastic Load Balancing, and Amazon CloudWatch
- <u>Using Public Data Sets</u>
- Reserving Amazon EC2 Instances
- Using Amazon Virtual Private Cloud

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



For detailed information the command line tools, go to the <u>Amazon Elastic Compute Cloud Command Line Reference</u>. For detailed information about the APIs, go to the <u>Amazon Elastic Compute Cloud API Reference</u>

Using Credentials

Topics

•

How to Log In with Your Amazon Login and Password

- How to Get Your Access Key ID and Secret Access Key
- How to Create an X.509 Certificate and Private Key
- SSH Key Pair
- Windows Administrator Password
- Viewing Your Account ID
- Related Topics

This section describes how to use Amazon EC2 credentials.

Amazon EC2 Credentials

- **Amazon Login and Password**—Used to sign up for Amazon EC2 and other services, view your bills, perform account-based tasks, and get many of your security credentials. Additionally, they are used by the AWS Management Console. For information on how to log in with your Amazon login and password, see <u>SSH Key Pair</u>.
- Access Key ID and Secret Access Key—Used most commonly used set of AWS credentials, they are used to make Query and REST-based requests and are commonly used by UI-based tools, such as ElasticFox. For more information, see Windows Administrator Password.
- **X.509 Certificate and Private Key**—The X.509 Certificate and Private Key are used by the command line tools and SOAP. For more information, see Windows Administrator Password.
- Access Key ID and Secret Access Key—Used to access instances after they launch through SSH and to generate passwords used by Remote Desktop. For more information, see <u>SSH Key Pair</u>.
- **EC2 (SSH) Key pair**—Used to access Windows instances through Remote Desktop. For more information, see <u>Windows Administrator Password</u>.

• **Account ID**— For more information, see <u>Windows Administrator</u> <u>Password</u>.

How to Log In with Your Amazon Login and Password

The Amazon login and password enable you to sign up for services, view your bills, perform account-based tasks, and get many of your security credentials. These credentials are also used to perform Amazon EC2 tasks through the AWS Management Console.

This section describes how to get your Amazon login and password.

To get your Amazon login and password (if you have an existing account)

- 1. Go to the AWS Web Site.
- 2. Select an option from the Your Account menu. The Amazon Web Services Sign In page appears.
- 3. Enter your e-mail address, select I am a returning user and my password is, enter your password, and click the Sign In button.

To get a new Amazon login and password

- 1. Go to the AWS Web Site.
- 2. Click Create an AWS Account.

The Amazon Web Services Sign In page appears.

- 3. Select I am a new user and click the Sign In button.
- 4. Follow the on-screen prompts to create a new account.



It is important to keep your Amazon login and password secret as they can be used to view and create new credentials. As an increased security measure, we offer Multi-Factor Authentication, which uses the combination of a physical device and passcode to login to your AWS account. For more information, go to http://aws.amazon.com/mfa.

How to Get Your Access Key ID and Secret Access Key

The Access Key ID and Secret Access Key are the most commonly used set of AWS credentials. They are used to make Query and REST-based requests and are commonly used by UI-based tools, such as ElasticFox. You can use up to two sets of Access Keys at a time. You can generate new keys at any time or disable existing keys.

To get your Access Key ID and Secret Access Key

- 1. Go to the AWS Web Site.
- 2. Point to Your Account and select Security Credentials.

If you are not already logged in, you are prompted to do so.

- 3. Scroll down to the Access Credentials section and verify the Access Keys tab is selected.
- 4. Locate an active Access Key in the Your Access Keys list.
- 5. To display the Secret Access Key, click Show in the Secret Access Key column.
- 6. Write down the keys or save them.
- 7. If there are no Access Keys in the list, click Create a New Access Key and follow the on-screen prompts.

How to Create an X.509 Certificate and Private Key

The X.509 Certificate and Private Key are used by the command line tools and SOAP. You can download the private key file once. If you lose it, you will need to create a new certificate. Up to two certificates can be active at any time.

This section describes how to create a new certificate.

To create a certificate

- 1. Go to the AWS Web Site.
- 2. Point to Your Account and select Security Credentials.

If you are not already logged in, you are prompted to do so.

- 3. Click the X.509 Certificates tab
- 4. Click Create a New Certificate and follow the on-screen prompts.
- 5. To display the Secret Access Key, click Show in the Secret Access Key column.

The new Certificate is created and appears in the X.509 Certificate list. You are prompted to download the certificate and private key files.

6. Create a .ec2 directory in your home directory, and save these files to it with the filenames offered by your browser.

You should end up with a PEM-encoded X.509 certificate and a private key file named as shown in the following examples. The following is an example of a PEM encoded signed X.509 certificate file: cert-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CL0.pem. The following is an example of an unencrypted, PEM encoded RSA private key file that corresponds to the X.509 certificate file: pk-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CL0.pem.



acceptable, but C:\My Documents\EC2 is not.

SSH 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.

For information on how to get the create key pairs, see <u>How to Generate an SSH Key Pair</u>.

Windows Administrator Password

The Windows administrator password is used to access a Windows instance through Remote Desktop for the first time only. If you change the password or rebundle the AMI, the instance will use the last set password. For information on how to get the Windows administrator password through the command line tools or the AWS Management Console, see ???.

Viewing Your Account ID

The Account ID identifies your account to AWS and enables other accounts to access resources that you want to share, such as Amazon EC2 AMIs and Amazon EBS snapshots.

To view your Account ID

- 1. Go to the AWS Web Site.
- 2. Point to Your Account and select Security Credentials.

If you are not already logged in, you are prompted to do so.

- 3. Scroll down to the section that contains information about your account.
- 4. Click Create a New Certificate and follow the on-screen prompts.
- 5. Locate your AWS Account ID.

For information on how to share AMIs, see <u>How to Share AMIs</u>. For information on how to share snapshots, see <u>How to Modify Snapshot Permissions</u>.



Note

The Account ID number is not a secret. When granting access to resources, make sure to specify the Account ID without hyphens.

The canonical ID is used by Amazon S3.

Related Topics

• <u>Amazon EC2 Credential Concepts</u>

Preparing and Creating AMIs

Topics

- Preparing an AMI
- Creating an AMI
- How to Share AMIs
- Creating Paid AMIs

This section describes how to prepare and create, and share Amazon Machine Images (AMIs).

Preparing an AMI

Topics

Preparing a Linux or UNIX AMIPreparing a Windows AMI

Preparing a Linux or UNIX AMI

Topics

- Starting with an Existing AMI
- Preparing an AMI through a Loopback File

In Linux and UNIX, there are two common ways to prepare 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. This is applicable for both Amazon EBS-backed and Amazon S3-backed AMIs, and is described in <u>Starting with an Existing AMI</u>.

Another approach is to build a fresh installation either on a stand-alone machine or on an empty file system mounted by loopback. This is only applicable for AMIs backed by Amazon S3 and entails building an operating system installation from scratch. For more information, see Preparing an AMI through a Loopback File.

After the installation package has been built to your satisfaction, you must create and register it as described in <u>Creating an AMI</u>.



Note

Preparing an AMI backed by Amazon S3 requires you to download and install the AMI tools in addition to the API tools. The AMI tools are not required for AMIs that use Amazon EBS volumes as their root devices. For more information on AMI tools, refer to the <u>Amazon Elastic Compute Cloud Getting Started Guide</u>.

This section provides detailed instructions on preparing an AMI. For information on quickly launching an existing AMI, go to the <u>Amazon Elastic Compute Cloud Getting Started Guide</u>.

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 as described in Creating an AMI.



Before selecting an AMI, determine whether the instance types you plan to launch are 32-bit or 64-bit. For more information, see <u>Instance Types</u>

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
2	How to Generate a Key Pair
3	How to Launch the Instance
4	How to Authorize Network Access
5	How to Connect to the Instance
6	How to Upload the Key and Certificate

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 475219
IMAGE ami-61a54028 <your-s3-bucket>/image.manifest.xml AIDADH4IGTRXXKCD
IMAGE ami-2bb65342 ec2-public-images/getting-started.manifest.xml 475219
IMAGE ami-6ea54007 ec2-public-images/base-fc3-mysql.manifest.xml 4752198
```

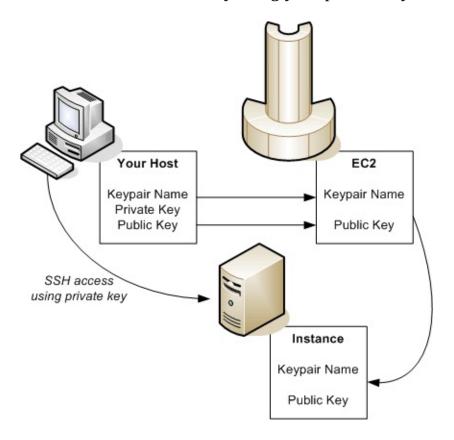


If you are planning on creating an AMI that uses an Amazon EBS volume as its root device, we recommend selecting an existing AMI backed by Amazon EBS and modifying it to meet your needs. For more information, see AMI and Instance Concepts

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.



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:1
----BEGIN RSA PRIVATE KEY----
MIIEoQIBAAKCAQBuLFq5ujHrtm1jnutSuo08Xe56LlT+HM8v/xkaa39EstM3/aFxTHqElQi.
HungXQ29VTc8rc1bW0lkdi230H5eqkMHGhvEwqa0HWASUMll4o3o/IX+0f2UcPoKC0VUR+j>
5AU52EQfanIn3ZQ8lFW7Edp5a3q4DhjGlUKToHVbicL5E+g45zfB95wIyywWZfeW/UUF3Lp(
ebIUla1aTbHkLbCC2r7RTn8vp0Wp47BGVYGtGSBMpTRP5hnbzzuai3itkiLHiU39S2sJCJ07
i8BygR4s3mHKBj8l+ePQxG1kGbF6R4yq6sECmXn17MRQVXODNHZbAqMBAAECqqEAY1tsiUs]
91CXirkYGuVfLyLflXenxfI50mDFms/mumTqloH07tr0oriHDR5K7wMcY/YY5YkcXNo7mvU\
ZNUJs7rw9gZRTrf7LylaJ58k0cyajw8TsC4e4LPbFaHwS1d6K8rXh64o6WgW4SrsB6ICmr1l
3wcfgt5ecIu4TZf00E9IHjn+2eRlsrjBde0Ri7KiUNC/pAG23I6MdD0FEQRcCSigCj+4/mci
SWS4dMbrpb9FNSIcf9dcLxVM7/6KxgJNfZc9XWzUw77Jg8x92Zd0fVhH0ux5IZC+UvSKWB4c
tE8C3p9bbU9VGyY5vLCAiIb4qQKBgQDLi024GXrIkswF32YtBBMuVgLGCwU9h9Hl09mKAc2r
jUE5IpzRjTedc9I2qiIMUTwtqnw42auSCzbUeYMURPtDqyQ7p6AjMujp9EPemcSV0K9vXYL(
xW9MC0dtV6iPkCN7g0giZXPRKaFbWADp16p8UAIvS/a5XXk5jwKBgQCKkpHi2EISh1uRkhx1
iDCiK6JBRsMvpLbc0v5dKwP5alo1fmdR5PJaV2qvZSj5CYNpMAv1/EDNTY50SIJU+0KFm0bv
rdLNLDL4+TcnT7c62/aH01ohYaf/VCbRhtLlBfgGoQc7+sAc8vmKkesnF7CgCEKDyF/dhrx\
gC0iZzzNAapayz1+JcVTwwEid6j9JgNXbBc+Z2YwMi+T0Fv/P/hwkX/ype0XnIUcw0Ih/Yt(
DQbsz7LcY1HqXiHKYNWNvXgww0+oiChjxvEkSdsTTIfnK4VSCvU9BxDbQHjdiNDJbL6oar92
rBYvChJZF7LvUH4YmVpHAoGAbZ2X7XvoeE0+uZ58/BGK0IGHByHBDiXtzMhdJr15HTYjxK7(
gK+8zp4L9IbvLGDMJ08vft32XPEWuvI8twCzFH+CsWLQADZMZKSsBas0Z/h1FwhdMgCMcY+(
JZKjTSu3i7vhvx6RzdSedXEMNTZWN4qlIx3kR5aHcukCqYA9T+Zrvm1F0seQPbLknn7EqhX]
P8TTvW/6bdPi23ExzxZn7KOdrfclYRph1LHMpAONv/x2xALIf91UB+v5ohy1oDoasL0gij1k
2ERKKdwz0ZL9SWg6VTdhr/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 stop (Amazon EBS-backed instances only) or 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
```

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
```

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_addre
```

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.



For information on obtaining your Amazon EC2 private key and X.509 certificate, refer to the <u>Amazon Elastic Compute Cloud Getting Started Guide</u>.

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.

Example

```
$ scp pk-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CL0.pem
cert-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CL0.pem
root@ec2-67-202-51-223.compute-1.amazonaws.com:/mnt
-i id_rsa-gsg-keypair
pk-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CL0.pem 100% 717 0.7KB/s 00:00
cert-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CL0.pem
```

100% 685 0.7KB/s 00:00

How to Load Software and Make Changes

Now that you are logged into the instance, you can load software and make changes as you would with any Linux or UNIX server. When you are finished with your changes, you can create a new AMI and launch an identical copy at any time. For more information AMIs, see Creating an AMI

Preparing 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. For more information on the AMI tools, go to the Amazon Elastic Compute Cloud Command Line Reference.



This method only works with AMIs that use instance stores for their root devices. This method is not applicable for AMIs backed by Amazon EBS.

Before selecting an AMI, determine whether the instance types you plan to launch are 32-bit or 64-bit. For more information, see <u>Instance Types</u>

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

How to Create a File to Host the AMI.
 How to Create a Root File System inside the File.
 How to Mount the File through Loopback.
 How to Prepare for the Installation.
 How to Install the Operating System.
 How to Configure the Operating System.

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

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-$r
enabled=1

[updates-released]
name=Fedora Core 4 - $basearch - Released Updates
mirrorlist=http://fedora.redhat.com/download/mirrors/updates-releas
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
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
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=<<u>image_mountpoint</u>> -y groupins
```

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% |=======|
                                                00
updates-released
                  100% |========|
                                         1.1 kB
                                                00
comps.xml
                  100% |====== | 693 kB
                                                00
comps.xml
                  100% |====== | 693 kB
                                                00
Setting up repositories
Reading repository metadata in from local files
                  100% |====== | 824 kB
                                                00
primary.xml.gz
       Added 2772 new packages, deleted 0 old in 15.32 seconds
primary.xml.gz
                  100% |======= | 824 kB
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 B00TPROT0=dhcp ONB00T=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:

|--|



The /dev/sda2 and /dev/sda3 storage locations only apply to small instances. For more information on instance storage, see Instance Storage.

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 myservice 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
```

Preparing a Windows AMI

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



Before selecting an AMI, determine whether the instance types you plan to launch are 32-bit or 64-bit. For more information, see <u>Instance Types</u>

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

Tasks to Use an Existing AMI

1	How to Select an AMI
2	How to Generate a Key Pair
3	How to Launch the Instance
4	How to Get the Administrator Password
5	How to Authorize Network Access
6	How to Connect to the Instance
7	How to Load Software and Make Changes

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

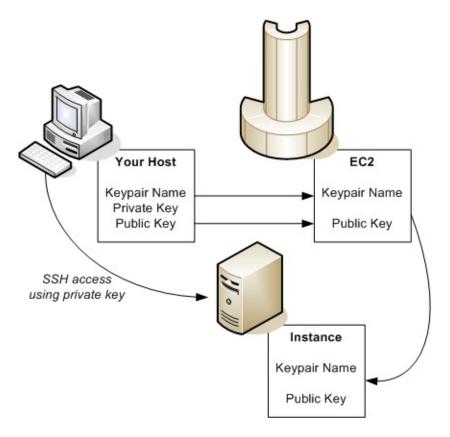
IMAGE ami-e6cd298f Key2047/test.manifest.xml 720208972571 available
```

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.



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:1 ----BEGIN RSA PRIVATE KEY----MIIEoQIBAAKCAQBuLFg5ujHrtm1jnutSuo08Xe56LlT+HM8v/xkaa39EstM3/aFxTHgElQi. HungXQ29VTc8rc1bW0lkdi230H5eqkMHGhvEwqa0HWASUMll4o3o/IX+0f2UcPoKC0VUR+j> 5AU52EQfanIn3ZQ8lFW7Edp5a3g4DhjGlUKToHVbicL5E+g45zfB95wIyywWZfeW/UUF3Lp(ebIUlq1qTbHkLbCC2r7RTn8vpQWp47BGVYGtGSBMpTRP5hnbzzuqj3itkiLHjU39S2sJCJ07 i8BygR4s3mHKBj8l+ePQxG1kGbF6R4yg6sECmXn17MRQVX0DNHZbAgMBAAECggEAY1tsiUs] 91CXirkYGuVfLyLflXenxfI50mDFms/mumTqloH07tr0oriHDR5K7wMcY/YY5YkcXNo7mvU\ ZNUJs7rw9gZRTrf7LylaJ58k0cyajw8TsC4e4LPbFaHwS1d6K8rXh64o6WgW4SrsB6ICmr1 3wcfgt5ecIu4TZf00E9IHjn+2eRlsrjBde0Ri7KiUNC/pAG23I6MdD0FEQRcCSigCj+4/mci SWS4dMbrpb9FNSIcf9dcLxVM7/6KxqJNfZc9XWzUw77Jq8x92Zd0fVhH0ux5IZC+UvSKWB4c tE8C3p9bbU9VGyY5vLCAiIb4qQKBgQDLi024GXrIkswF32YtBBMuVgLGCwU9h9Hl09mKAc2r jUE5IpzRjTedc9I2qiIMUTwtgnw42auSCzbUeYMURPtDqyQ7p6AjMujp9EPemcSV0K9vXYL(xW9MC0dtV6iPkCN7q0qiZXPRKaFbWADp16p8UAIvS/a5XXk5jwKBqQCKkpHi2EISh1uRkhx1 iDCiK6JBRsMvpLbc0v5dKwP5alo1fmdR5PJaV2qvZSj5CYNpMAy1/EDNTY50SIJU+0KFmQby rdLNLDL4+TcnT7c62/aH01ohYaf/VCbRhtLlBfgGoQc7+sAc8vmKkesnF7CgCEKDyF/dhrx\ gC0iZzzNAapayz1+JcVTwwEid6j9JqNXbBc+Z2YwMi+T0Fv/P/hwkX/ype0XnIUcw0Ih/Yt(DQbsz7LcY1HqXiHKYNWNvXqww0+oiChjxvEkSdsTTIfnK4VSCvU9BxDbQHjdiNDJbL6oar92 rBYvChJZF7LvUH4YmVpHAoGAbZ2X7XvoeE0+uZ58/BGK0IGHByHBDiXtzMhdJr15HTYjxK7(qK+8zp4L9IbvLGDMJ08vft32XPEWuvI8twCzFH+CsWL0ADZMZKSsBas0Z/h1FwhdMqCMcY+(JZKjTSu3i7vhvx6RzdSedXEMNTZWN4qlIx3kR5aHcukCgYA9T+Zrvm1F0seQPbLknn7EqhX] P8TTvW/6bdPi23ExzxZn7KOdrfclYRph1LHMpAONv/x2xALIf91UB+v5ohy1oDoasL0gij1k 2ERKKdwz0ZL9SWg6VTdhr/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 stop (Amazon EBS-backed instances only) or 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

The following shows the status of the launch:

How to Get the Administrator Password

After you launch an instance, get its administrator password.



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

Qr89fdS1w
```

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:

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 defau
INSTANCE i-ae0bf0c7 ami-2bb65342 ec2-67-202-7-236
```

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.

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 Creating an AMI



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

Creating an AMI

Topics

- Creating an Amazon EBS-Backed AMI
- Creating an Amazon S3-Based AMI

This section describes how to create your own AMI from the AMI that you prepared. For information on preparing an AMI, see Preparing an AMI.

Creating an Amazon EBS-Backed AMI

Topics

•

How To

This section describes how to create an AMI that uses an Amazon EBS volume as its root device, which is easier to create than AMIs that use instance stores as their root devices. More importantly, instances launched from these AMIs launch faster and use persistent storage. For more information, see AMI and Instance Concepts.



If you are creating an AMI that uses an instance store as its root device, see .<u>Creating an AMI</u>.

Amazon EBS-backed instances are stored as Amazon EBS snapshots. Standard storage rates apply.

This procedure is applicable for both Amazon EBS-backed Linux/UNIX and Windows AMIs.

Sysprep does not automatically run for Amazon EBS-backed Windows instances,

How To

To create an Amazon EBS-Backed AMI

1. If you are using SOAP, construct the following request:

```
<CreateImage xmlns="http://ec2.amazonaws.com/doc/2009-10-31/">
     <InstanceId>instance_id</InstanceId>
     </CreateImage>
```

2. If you are using Query, construct the following request:

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

3. View output similar to the following:

Creating an Amazon S3-Based AMI

Topics

•

Bundling a Linux or UNIX AMI

• Bundling a Windows AMI

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



Note

During bundling, only the root store (e.g., / or C:\) is bundled. Data on other instance stores is not preserved. For more information, see <u>AMI and Instance Concepts</u>

Bundling a Linux or UNIX AMI

The AMI tools include three command line utilities:

- ec2-bundle-image bundles an existing AMI
- ec2-bundle-vol 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 <u>public Amazon S3 downloads bucket</u>. 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 preparing 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 preparing AMIs, see Preparing an AMI).

To bundle the loopback file image

• Enter the following command:

```
# ec2-bundle-image -i <image_name>.img -k /private_keyfile> -C <certificate_file>
```

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.



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-HKZ
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-HKZYKTAIG2ECMXYIBH3HX 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 <sec
```

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

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



If you make any changes to the source image stored in Amazon S3, you must reregister 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 <u>Windows Configuration Service</u>

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.

2. If you want to reduce your startup time, delete any temporary files on your instance using the Disk Cleanup tool, defragment your system using Disk Defragmenter, and zero out free space using sdelete -c C:\.



The sdelete utility is available from the sdelete Download Page or the

Microsoft Web Site.

3. 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> -0 <a
```

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 eVBUNDLE bun-e3a4418a i-eb977f82 mybucket myimage 2008-10-02T09:31:44+0000 2
```

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-
```

How to Register the AMI

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



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

• Sharing AMIs

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 <u>Protecting a Shared AMI (Linux and UNIX)</u>. For information on sharing AMIs, see <u>Sharing AMIs</u>

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
2	Disable Password-Based Logins for Root
3	Install Public Key Credentials
4	How to Disable sshd DNS Checks (optional)
5	Identify Yourself
6	Protect Yourself
7	Protect Paid AMIs



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 <u>AWS forums</u> in case someone else has done it already.

How to Update the AMI Tools at Boot Time

For AMIs backed by Amazon S3, 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
```



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.



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.

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



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.



Before proceeding, make sure to read the security considerations of sharing AMIs in the <u>Protecting a Shared AMI (Linux and UNIX)</u> 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
- How to Share an AMI with Specific Users
- How to Publish Shared AMIs

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.



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 AIDADH4IGTRXXKCI
```

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 AIDADH4IGTRXXKCI
```

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 <u>Amazon EC2</u> Resource Center.

To publish your AMI

- 1. Post it in the Public AMIs Folder of the <u>Amazon Web Services Resource</u> <u>Center</u>, 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.

```
| Compared to the state of 
                                                       This image contains the following daemons / services:
                                                       ul>
                                                                                                                  | Compared to the compared
                                                                                                                  | Compared to the compared
                                                       <h2><strong>What&#39;s New?</strong></h2>The following changes were
ul>
                                                       [Release Notes 1]
<span style="font-size: x-small; font-family: courier new,courier">
<span style="font-size: x-small; font-family: courier new,courier">
<span style="font-size: x-small; font-family: courier new,courier">
<l
```

Creating Amazon EBS-Backed Shared AMIs

This section describes how to create a public AMI backed by Amazon EBS.

First Time

This section describes how to create a public AMI backed by Amazon EBS for the first time.

To create an AMI

- 1. Launch an Amazon S3-backed instance. See <u>Launching and Using</u> <u>Instances</u>.
- 2. Create an Amazon EBS volume, specifying a size that meets your needs. See <u>How to Create an Amazon EBS Volume</u>.
- 3. Attach the volume to your instance. See <u>How to Attach the Volume to an</u> <u>Instance</u>.
- 4. Partition and format the volume. Then, copy the operating system. See <u>How</u> to Make an Amazon EBS Volume Available for Use.
- 5. Mount the volume. See <u>How to Make an Amazon EBS Volume Available</u> for Use.
- 6. Configure the instance to meet your needs.
- 7. Unmount the volume. See <u>How to Detach an Amazon EBS Volume from an Instance</u>.
- 8. Create a snapshot of the new volume. See <u>How to Create an Amazon EBS Snapshot</u>.
- 9. Register the snapshot as an AMI. See <u>How to Register the AMI</u>.

Subsequent Times

This section describes how to create a public AMI backed by Amazon EBS, based on previously created AMIs.

To create an AMI

- 1. Launch an Amazon S3-backed instance. See <u>Launching and Using</u> Instances.
- 2. Create a Amazon EBS volume from a previously created snapshot. See How to Create an Amazon EBS Volume.
- 3. Mount the volume. See <u>How to Make an Amazon EBS Volume Available</u> for Use.
- 4. Configure the instance to meet your needs.
- 5. Unmount the volume. See <u>How to Detach an Amazon EBS Volume from an Instance</u>.
- 6. Create a snapshot of the new volume. See <u>How to Create an Amazon EBS</u> <u>Snapshot</u>.
- 7. Register the snapshot as an AMI. See <u>How to Register the AMI</u>.

Creating Paid AMIs

Topics

•

Amazon DevPay and Paid AMIs

- Product Registration
- How to Associate a Product Code with an AMI
- How to Share Your Paid AMI
- How to Confirm an Instance Is Running with a Product Code
- How to Get the Product Code from Within an Instance
- Supported AMIs

This section gives an introduction to Amazon DevPay and paid AMIs, which are AMIs you sell to other Amazon EC2 users.

Amazon DevPay and Paid AMIs

A *paid AMI* is an AMI that you sell to other Amazon EC2 users. They pay you according to the price you set. To be able to create a paid AMI, you use Amazon DevPay. What is Amazon DevPay?

Amazon DevPay is a billing and account management service that enables you to get paid for an AMI you create and that other Amazon EC2 users use. Amazon DevPay creates and manages the order pipeline and billing system for you. Your customers sign up for your AMI, and Amazon DevPay automatically meters their usage of Amazon EC2, bills them based on the pricing you set, and collects their payments. DevPay offers the following:

- You can charge customers for your AMI; the charges can include recurring charges based on the customer's usage Amazon EC2, a fixed one-time charge, and a recurring monthly charge.
- Your customers can easily sign up and pay for your AMI with their trusted Amazon.com accounts.
- Your customers are authenticated, thus ensuring they have access only to what they should.
- If your customers don't pay their bills, DevPay turns off their access to your AMI for you.
- Amazon Payments handles payment processing.



Basic DevPay Flow

	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.

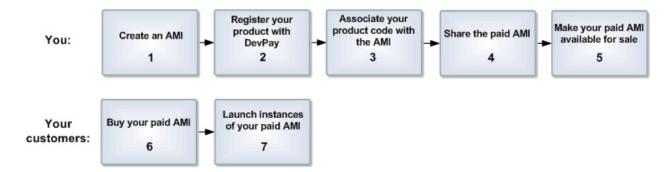
You can also use Amazon EC2 and Amazon DevPay together with a *supported AMI*. For more information about supported AMIs, see <u>Supported AMIs</u>.

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 <u>Product Registration</u>. 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 *purchase URL*).

3 You use an Amazon EC2 command or API call to associate the product code with your AMI.

For more information, see <u>How to Associate a Product Code with an AMI</u>. 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**.



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 <u>Solutions Catalog</u> on the AWS Developer Connection site and on the <u>Amazon Machine Images (AMIs)</u> 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 <u>Paying for AMIs</u>.



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.

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 <u>Creating an AMI</u>). 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.

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.

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 <u>Creating an AMI</u>). 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 <u>Summary of How Paid AMIs Work</u> and <u>Supported AMIs</u>. 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*.



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:

The code is the product code and <instance</pre> 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 has an Amazon DevPay product code. The instance retrieves the product code similarly to how retrieves other metadata. For more information about retrieving metadata, see Instance Metadata.

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:

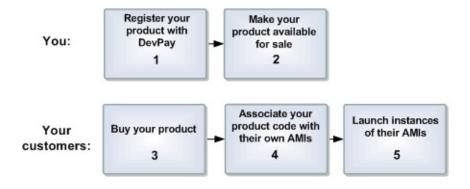
- Paid AMI—You associate your own product code with your own AMI
- Supported AMI—Other EC2 users associate your product code with their own AMIs



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.

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 <u>Product Registration</u>. 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 *purchase URL*).

- 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**.

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.

Launching and Using Instances

Topics

•

How to Find a Suitable AMI

- How to Generate an SSH Key Pair
- How to Add Rules to the Default Security Group
- How to Run an Instance
- How to Start and Stop Instances
- Instance Metadata
- Instance Storage Usage
- Using Shared AMIs
- Paying for AMIs
- Getting Console Output and Rebooting Instances
- Related Topics

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.



If you create an instance in one Region, you cannot launch it in another Region without migrating it. For information on Regions, see <u>Region and Availability</u> <u>Zone Concepts</u>. For information on migrating AMIs, refer to the ec2-migrate-bundle section in the <u>Amazon Elastic Compute Cloud Command Line Reference</u>.

Related Topics

- Amazon EC2 Flow
- AMI and Instance Concepts

How to Find a Suitable AMI

This section describes how to find an AMI.

How To

To find a suitable AMI

1. If you are using SOAP, construct a request similar to the following which returns all Amazon-owned AMIs:

2. If you are using Query, construct a request similar to the following which returns all Amazon-owned AMIs:

```
https://ec2.amazonaws.com/
?Action=DescribeImages
&User.1=amazon
&...auth parameters...
```

3. View output similar to the following:

```
<kernelId>aki-d3376696/kernelId>
   <ramdiskId>ari-e73766a2/ramdiskId>
   <imageOwnerAlias>amazon</imageOwnerAlias>
   <name>getting-started</name>
   <description>Fedora 8 v1.11 i386 lvm-rootVG-rootFS ext3 ec2pn
   <rootDeviceType>ebs</rootDeviceType>
   <rootDeviceName>/dev/sda</rootDeviceName>
   <blockDeviceMapping>
      <item>
        <deviceName>/dev/sda</deviceName>
        <ebs>
          <snapshotId>snap-32885f5a</snapshotId>
          <volumeSize>15</volumeSize>
          <deleteOnTermination>false</deleteOnTermination>
        </ebs>
      </item>
   </blockDeviceMapping>
 </item>
</imagesSet>
```

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.



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 <u>Amazon Elastic Compute Cloud Getting Started Guide</u>.

How To

To generate a key pair

1. If you are using SOAP, construct the following request:

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=CreateVolume
&KeyName=gsg-keypair
&...auth parameters...
```

3. View output similar to the following:

```
<CreateKeyPairResponse xmlns="http://ec2.amazonaws.com/doc/2009-10-</pre>
  <keyName>gsg-keypair</keyName>
  <keyFingerprint>1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9
  <keyMaterial>----BEGIN RSA PRIVATE KEY-----
MIIEoQIBAAKCAQBuLFq5ujHrtm1jnutSuo08Xe56LlT+HM8v/xkaa39EstM3/aFxTHq
HungXQ29VTc8rc1bW0lkdi230H5egkMHGhvEwga0HWASUMll4o3o/IX+0f2UcPoKCOV
5AU52EQfanIn3ZQ8lFW7Edp5a3q4DhjGlUKToHVbicL5E+g45zfB95wIyywWZfeW/UU
ebIUlq1qTbHkLbCC2r7RTn8vpQWp47BGVYGtGSBMpTRP5hnbzzuqj3itkiLHjU39S2s
i8BygR4s3mHKBj8l+ePQxG1kGbF6R4yg6sECmXn17MRQVX0DNHZbAgMBAAECggEAY1t
91CXirkYGuVfLyLflXenxfI50mDFms/mumTqloH07tr0oriHDR5K7wMcY/YY5YkcXNo
ZNUJs7rw9gZRTrf7LylaJ58k0cyajw8TsC4e4LPbFaHwS1d6K8rXh64o6WgW4SrsB6I
3wcfqt5ecIu4TZf00E9IHin+2eRlsrjBde0Ri7KiUNC/pAG23I6MdD0FEQRcCSiqCj+
SWS4dMbrpb9FNSIcf9dcLxVM7/6KxgJNfZc9XWzUw77Jg8x92Zd0fVhH0ux5IZC+UvS
tE8C3p9bbU9VGyY5vLCAiIb4qQKBqQDLi024GXrIkswF32YtBBMuVgLGCwU9h9Hl09m
jUE5IpzRjTedc9I2qiIMUTwtgnw42auSCzbUeYMURPtDqyQ7p6AjMujp9EPemcSV0K9
xW9MC0dtV6iPkCN7g0giZXPRKaFbWADp16p8UAIvS/a5XXk5jwKBgQCKkpHi2EISh1u
iDCiK6JBRsMvpLbc0v5dKwP5alo1fmdR5PJaV2qvZSj5CYNpMAy1/EDNTY50SIJU+0K
rdLNLDL4+TcnT7c62/aH01ohYaf/VCbRhtLlBfqGoQc7+sAc8vmKkesnF7CqCEKDyF/
gC0iZzzNAapayz1+JcVTwwEid6j9JgNXbBc+Z2YwMi+T0Fv/P/hwkX/ype0XnIUcw0I
DQbsz7LcY1HqXiHKYNWNvXgww0+oiChjxvEkSdsTTIfnK4VSCvU9BxDbQHjdiNDJbL6
rBYvChJZF7LvUH4YmVpHAoGAbZ2X7XvoeE0+uZ58/BGK0IGHByHBDiXtzMhdJr15HTY
gK+8zp4L9IbvLGDMJ08vft32XPEWuvI8twCzFH+CsWLQADZMZKSsBas0Z/h1FwhdMgC
```

JZKjTSu3i7vhvx6RzdSedXEMNTZWN4qlIx3kR5aHcukCgYA9T+Zrvm1F0seQPbLknn7P8TTvW/6bdPi23ExzxZn7KOdrfclYRph1LHMpAONv/x2xALIf91UB+v5ohy1oDoasL02ERKKdwz0ZL9SWq6VTdhr/5G994CK72fy5WhyERbDjUIdHaK3M849JJuf8cSrvSb4g=----END RSA PRIVATE KEY----</keyMaterial></CreateKeyPairResponse>

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.

How To

To authorize access to your instance

1. If you are using SOAP, construct the following request:

```
<AuthorizeSecurityGroupIngress xmlns="http://ec2.amazonaws.com/doc/</pre>
<! [CDATA[
    <userId/>
    <groupName>default
    <ipPermissions>
        <item>
            <ipProtocol>tcp</ipProtocol>
            <fromPort>80</fromPort>
            <toPort>80</toPort>
            <groups/>
            <ipRanges>
                <item>
                     <cidrIp>0.0.0.0/0</cidrIp>
                </item>
            </ipRanges>
        </item>
        <item>
            <ipProtocol>tcp</ipProtocol>
            <fromPort>3389</fromPort>
            <toPort>3389</toPort>
            <groups/>
            <ipRanges>
                     <cidrIp>your-public-ip-address/32</cidrIp>
                </item>
            </ipRanges>
        </item>
        <item>
            <ipProtocol>tcp</ipProtocol>
            <fromPort>22</fromPort>
            <toPort>22</toPort>
            <groups/>
            <ipRanges>
                <item>
                     <cidrIp>your-public-ip-address/32</cidrIp>
                </item>
            </ipRanges>
        </item>
    </ipPermissions>
```

```
</AuthorizeSecurityGroupIngress>
```

2. If you are using Query, construct the following requests:

```
https://ec2.amazonaws.com/
?Action=AuthorizeSecurityGroupIngress
&GroupName=default
&IpProtocol=tcp
&FromPort=80
&ToPort=80
&CidrIp=0.0.0.0/0
&...auth parameters...
https://ec2.amazonaws.com/
?Action=AuthorizeSecurityGroupIngress
&GroupName=default
&IpProtocol=tcp
&FromPort=3389
&ToPort=3389
&CidrIp=your-public-ip-address/32
&...auth parameters...
https://ec2.amazonaws.com/
?Action=AuthorizeSecurityGroupIngress
&GroupName=default
&IpProtocol=tcp
&FromPort=22
&ToPort=22
&CidrIp=your-public-ip-address/32
&...auth parameters...
```

3. View output similar to the following:

How to Run an Instance

This section describes how to run an instance.

How To

To launch an instance

1. If you are using SOAP, construct the following request:

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=RunInstances
&ImageId=ami-235fba4a
&MaxCount=1
&MinCount=1
&KeyName=gsg-keypair
&Placement.AvailabilityZone=us-east-1a
&...auth parameters...
```

3. View output similar to the following:

```
<item>
      <instanceId>i-2ba64342</instanceId>
      <imageId>ami-235fba4a</imageId>
      <instanceState>
        <code>0</code>
        <name>pending</name>
      </instanceState>
      <privateDnsName></privateDnsName>
      <dnsName></dnsName>
      <keyName>gsg-keypair</keyName>
      <amiLaunchIndex>0</amiLaunchIndex>
      <instanceType>m1.small</instanceType>
      <launchTime>2007-08-07T11:51:50.000Z</launchTime>
      <placement>
        <availabilityZone>us-east-la</availabilityZone>
      </placement>
      <monitoring>
        <enabled>false</enabled>
      </monitoring>
    </item>
  </instancesSet>
</RunInstancesResponse>
```

To view the status of a launched instance

1. If you are using SOAP, construct the following request:

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=DescribeInstances
&InstanceId=i-2ba64342
&...auth parameters...
```

3. View output similar to the following:

```
<DescribeInstancesResponse xmlns="http://ec2.amazonaws.com/doc/2009</pre>
  <reservationId>r-47a5402e</reservationId>
 <ownerId>AIDADH4IGTRXXKCD</ownerId>
  <groupSet>
    <item>
      <groupId>default
    </item>
  </groupSet>
  <instancesSet>
    <item>
      <instanceId>i-2ba64342</instanceId>
      <imageId>ami-235fba4a</imageId>
      <instanceState>
        <code>0</code>
        <name>running</name>
      </instanceState>
      <privateDnsName></privateDnsName>
      <dnsName></dnsName>
      <keyName>gsg-keypair</keyName>
      <amiLaunchIndex>0</amiLaunchIndex>
      <instanceType>m1.small</instanceType>
      <launchTime>2007-08-07T11:51:50.000Z</launchTime>
      <placement>
        <availabilityZone>us-east-la</availabilityZone>
      </placement>
      <monitoring>
        <enabled>false/enabled>
      </monitoring>
    </item>
  </instancesSet>
</DescribeInstancesResponse>
```

How to Start and Stop Instances

This section describes how to start and stop instances that use Amazon EBS volumes as their root devices.

When an instance is stopped, it is saved as a snapshot, and you are not billed for hourly usage or data transfer. You can then start it at any time by using the start operation.



Before stopping an instance, make sure it is in a state from which it can be restarted. Stopping an instance does not preserve data stored in RAM.

Performing this operation on an instance that uses an instance store as its root device returns an error.

When you start a stopped instance, the IP address is likely to change. By default, the Ec2ConfigService service (Windows) changes the instance hostname to match the new IP and initiates a reboot.

How To

To stop an instance

1. If you are using SOAP, construct the following request:

```
<StopInstances xmlns="http://ec2.amazonaws.com/doc/2009-10-31/">
     <instanceId>i-10a64379</instanceId>
     </StopInstances>
```

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=StopInstances
&InstanceId=i-10a64379
&...auth parameters...
```

3. View output similar to the following:

To start an instance

1. If you are using SOAP, construct the following request:

```
<StartInstances xmlns="http://ec2.amazonaws.com/doc/2009-10-31/">
     <instanceId>i-10a64379</instanceId>
     </StartInstances>
```

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=StartInstances
&InstanceId=i-10a64379
```

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

3. View output similar to the following:

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.



For information on categories of metadata, see Metadata Categories.

Data Retrieval

An instance retrieves the data by querying a web server using a Query API. The base URI of all requests is http://169.254.169.254/2009-10-31/ where 2009-10-31 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 <u>Amazon Elastic Compute Cloud API</u> 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
2009-10-31
```

This example gets the top-level metadata items.

```
GET http://169.254.169.254/2009-10-31/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://169.254.169.254/2009-10-31/meta-data/ami-manifest-path my-amis/spamd-image.manifest.xml

GET http://169.254.169.254/2009-10-31/meta-data/ami-manifest-path my-amis/spamd-image.manifest.xml

GET http://169.254.169.254/2009-10-31/meta-data/ami-id ami-2bb65342

GET http://169.254.169.254/2009-10-31/meta-data/reservation-id r-fea54097

GET http://169.254.169.254/2009-10-31/meta-data/hostname ec2-67-202-51-223.compute-1.amazonaws.com
```

This example gets the list of available public keys.

```
GET http://169.254.169.254/2009-10-31/meta-data/public-keys/

0=my-public-key
```

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

```
GET http://169.254.169.254/2009-10-31/meta-data/public-keys/0/
openssh-key
```

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

```
GET http://169.254.169.254/2009-10-31/meta-data/public-keys/0/openssh-key ssh-rsa AAAA....wZEf my-public-key
```

This example gets the product code.

```
GET http://169.254.169.254/2009-10-31/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://169.254.169.254/2009-10-31/user-data 1234,fred,reboot,true | 4512,jimbo, | 173,,,
```

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

```
GET http://169.254.169.254/2009-10-31/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 | replicat
```

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-e RESERVATION r-fea54097 598916040194 default INSTANCE i-3ea74257 ami-2bb65342 pending 0 ml.small 2007-08-07T11:29:58-INSTANCE i-31a74258 ami-2bb65342 pending 1 ml.small 2007-08-07T11:29:58-INSTANCE i-31a74259 ami-2bb65342 pending 2 ml.small 2007-08-07T11:29:58-INSTANCE i-31a7425a ami-2bb65342 pending 3 ml.small 2007-08-07T11:29:58-INSTANCE i-31a7425a ami-2bb65342 pending 3 ml.small 2007-08-07T11:29:58-
```

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.
```

```
GET http://169.254.169.254/2009-10-31/meta-data/ami-launch-index 1
```

2 Retrieve the user data.

```
GET http://169.254.169.254/2009-10-31/user-data store-size=123PB backup-every=5min | replicate-every=1min
```

3 Extract the appropriate part of the user data.

```
user_data.split('|')[ami_launch_index]
```

Instance Storage Usage

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 or the instance is stopped, the data will be lost.

If you need a permanent storage solution, we recommend using Amazon EBS. For more information, see <u>Amazon Elastic Block Store</u>.



If you do not use Amazon EBS, make sure to back 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 onscreen prompts.



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:

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

```
\label{eq:dev/sde} \mbox{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 -0 <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.



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 ssh authorized keys 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 <u>Disable Password-Based Logins for Root</u>.
- 4 Check if ssh allows root password logins. See <u>Disable Password-Based Logins for Root</u> 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

- Purchasing a Paid AMI
- How to Launch Paid AMIs
- Paid Support
- Bills for Paid and Supported AMIs

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 <u>Amazon DevPay Developer Guide</u>. For more information about charging for your use of your AMIs, or providing support, see <u>Creating Paid AMIs</u>

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.



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:

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 7

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.

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.



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 Instances (Soncepts.)

To associate the product code with your AMI

• Enter the **ec2-modify-image-attribute** command:

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

The <ami_id> is the AMI ID and duct_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 runinstances. 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.



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.



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.



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

Accessing Instances

Topics

•

Accessing Instances in Linux and UNIX

- Accessing Instances in Windows
- Related Topics

This section describes how to access instances that you launched. For information on launching instances, see <u>Launching and Using Instances</u>

Related Topics

- Amazon EC2 Flow
- AMI and Instance Concepts

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_addre
```

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)
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.c 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.



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>
```

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:

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 defau
INSTANCE i-ae0bf0c7 ami-2bb65342 ec2-67-202-7-236
```

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 Windowsbased system.



Note

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

Using Instance Addressing

Topics

API Overview

- Determining Your IP AddressesUsing Elastic IP Addresses

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.

How To

To determine your private and public IP addresses

1. If you are using SOAP, construct the following request:

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=DescribeInstances
&InstanceId.1=instance-id
&...auth parameters...
```

```
<imageId>ami-6ea54007/imageId>
          <instanceState>
            <code>0</code>
            <name>running</name>
          </instanceState>
          <privateDnsName>10-251-50-132.ec2.internal</privateDnsNam</pre>
          <dnsName>ec2-72-44-33-4.compute-1.amazonaws.com</dnsName>
          <keyName>example-key-name</keyName>
          <amiLaunchIndex>23</amiLaunchIndex>
          cproductCodesSet>
            <item><productCode>774F4FF8</productCode></item>
          ductCodesSet>
          <instanceType>m1.large</instanceType>
          <launchTime>2007-08-07T11:54:42.000Z</launchTime>
          <placement>
            <availabilityZone>us-east-1b</availabilityZone>
          </placement>
          <kernelId>aki-ba3adfd3</kernelId>
          <ramdiskId>ari-badbad00/ramdiskId>
        </item>
      </instancesSet>
    </item>
  </reservationSet>
</DescribeInstancesResponse>
```

- 4. To determine the public IP address, use the IP address specified within dnsName.
- 5. To determine the private IP address, use the IP address specified within privateDnsName.

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.

How To

To allocate a new IP address for use with your account

1. If you are using SOAP, construct the following request:

```
<AllocateAddress xmlns="http://ec2.amazonaws.com/doc/2009-10-31/"/>
```

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=AllocateAddress
&...auth parameters...
```

3. View output similar to the following:

Describing Elastic IP Addresses

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

How To

To view elastic IP addresses assigned to your account

1. If you are using SOAP, construct the following request:

```
<DescribeAddresses xmlns="http://ec2.amazonaws.com/doc/2009-10-31/"</pre>
```

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=DescribeAddresses
&...auth parameters...
```

3. View output similar to the following:

Associating an Elastic IP Address with a Running Instance

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

How To

To associate an elastic IP address with an instance

1. Describe running instances and write down the instance ID of the instance

that you will associate with the elastic IP address. For more information, see How To.

- 2. Describe elastic IP addresses assigned to the account and write down the elastic IP address that you will associate with the instance. For more information, see How To.
- 3. Associate the instance and elastic IP address.

If you are using SOAP, construct the following request:

```
<AssociateAddress xmlns="http://ec2.amazonaws.com/doc/2009-10-31/">
     <instanceId>instance-id</instanceId>
        <publicIp>elastic-ip-address</publicIp>
</AssociateAddress>
```

If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=AssociateAddress
&InstanceId=<u>instance-id</u>
&PublicIp=<u>elastic-ip-address</u>
&...auth parameters...
```

Amazon EC2 returns output similar to the following:

4. Associations take a few minutes to complete. You can verify the association using DescribeAddresses or DescribeInstances.

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.



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

How To

To remap an IP address

- 1. Describe running instances and write down the instance ID of the instance that you will associate with the elastic IP address.
- 2. Describe elastic IP addresses assigned to the account and write down the elastic IP address to remap.
- 3. Associate the instance and elastic IP address.

If you are using SOAP, construct the following request:

```
<AssociateAddress xmlns="http://ec2.amazonaws.com/doc/2009-10-31/">
     <instanceId>instance-id</instanceId>
        <publicIp>elastic-ip-address</publicIp>
</AssociateAddress>
```

If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=AssociateAddress
&InstanceId=instance-id
&PublicIp=elastic-ip-address
&...auth parameters...
```

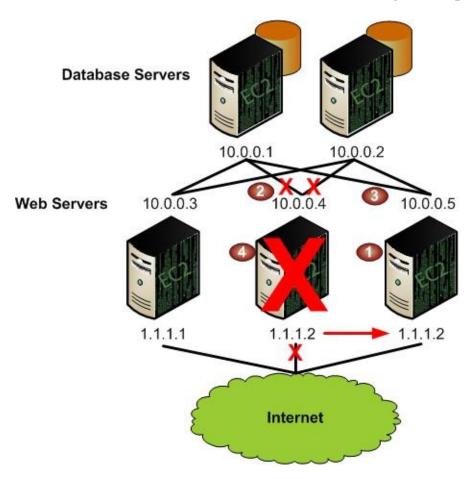
Amazon EC2 returns output similar to the following:

Associations take a few minutes to complete.

4. Verify the association using DescribeAddresses or DescribeInstances.

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).

Process

The following demonstrates how to set up these tasks using the API.

- 1. The user runs a new larger instance type using RunInstances.
- 2. The user disassociates the elastic IP address from the old instance using DisassociateAddress.



This step is optional as AssociateAddress automatically disassociates the elastic IP address if it is assigned to another instance.

- 3. The user associates the elastic IP address with the new instance using AssociateAddress.
- 4. The user verifies the association using DescribeInstances.
- 5. The user terminates the old instance using TerminateInstances.

Related Topics

• Instance Addressing Concepts

Using Network Security

Topics

•

API Overview

- Creating a Security Group
- Describing Security Groups
- Adding a Security Group Rule
- Delete a Security Group Rule
- Delete a Security Group
- Example

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.

How To

To create a security group

1. If you are using SOAP, construct the following request:

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action==CreateSecurityGroup
&GroupName=security-group-name
&GroupDescription=security-group-description
&...auth parameters...
```

```
<CreateSecurityGroupResponse xmlns="http://ec2.amazonaws.com/doc/20
  <return>true</return>
  </CreateSecurityGroupResponse>
```

Describing Security Groups

This section describes how to view currently configured security groups.

How To

To view security groups

1. If you are using SOAP, construct the following request:

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=DescribeSecurityGroups
&GroupName.1=security-group-name
&...auth parameters...
```

Adding a Security Group Rule

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

How To

To add a rule to a security group

1. If you are using SOAP, construct a request similar to the following:

```
<AuthorizeSecurityGroupIngress xmlns="http://ec2.amazonaws.com/doc/</pre>
    <userId/>
    <groupName>WebServers
    <ipPermissions>
        <item>
            <ipProtocol>tcp</ipProtocol>
            <fromPort>80</fromPort>
            <toPort>80</toPort>
            <groups/>
            <ipRanges>
                    <cidrIp>0.0.0.0/0</cidrIp>
                </item>
            </ipRanges>
        </item>
    </ipPermissions>
</AuthorizeSecurityGroupIngress>
```

2. If you are using Query, construct a request similar to the following:

```
https://ec2.amazonaws.com/
?Action=AuthorizeSecurityGroupIngress
&IpProtocol=tcp
&FromPort=80
&ToPort=80
&CidrIp=0.0.0.0/0
&...auth parameters...
```

<AuthorizeSecurityGroupIngressResponse xmlns="http://ec2.amazonaws.
 <return>true</return>
 </AuthorizeSecurityGroupIngressResponse>

Delete a Security Group Rule

This section describes how to delete a security group rule.

How To

To delete a security group rule

1. If you are using SOAP, construct a request similar to the following:

2. If you are using Query, construct a request similar to the following:

```
https://ec2.amazonaws.com/
?Action=RevokeSecurityGroupIngress
&IpProtocol=tcp
&FromPort=80
&ToPort=80
&CidrIp=0.0.0.0/0
&...auth parameters...
```

Delete a Security Group

This section describes how to delete a security group.

How To

To delete a security group

1. If you are using SOAP, construct the following request:

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=DeleteSecurityGroup
&GroupName=security-group-name
&...auth parameters...
```

```
<DeleteSecurityGroupResponse xmlns="http://ec2.amazonaws.com/doc/20
  <return>true</return>
  </DeleteSecurityGroupResponse>
```

Example

GROUP

PERMISSION

default

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

```
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
2 After a little wait for image launch to complete. Albert, who is a cautious type, checks the access rules
   PROMPT> ec2-describe-group default
            598916040194
                               default default group
   GROUP
   PERMISSION default ALLOWS all
                                            FROM
                                                     USER
                                                             598916040194
                                                                               GRPN
  Albert notices that it only accepts ingress network connections from other members of the default grou
3 Albert, being paranoid as well as cautious, uses the Linux and UNIX nmap command to port scan his ir
   s 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 ]
   All 100 scanned ports on ec2-67-202-51-105.compute-1.amazonaws.com
   Nmap finished: 1 IP address (1 host up) scanned in 31.008 seconds
 Albert decides he should be able to SSH into his instance, but only from his own machine.
```

PROMPT> ec2-authorize default -P tcp -p 22 -s 126.52.1.130/32

default ALLOWS tcp 22 22 FROM CIDR 126.52.1.130/3

Albert repeats the Linux and UNIX nmap port scan.

\$ 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]
Interesting ports on ec2-67-202-51-105.compute-1.amazonaws.com (67.
(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).

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

```
First, Mary creates a group for her Apache web server instances and allows HTTP access to the world
PROMPT> ec2-add-group apache -d "Mary's Apache group"
         apache Mary's Apache group
GROUP
PROMPT> ec2-describe-group apache
                           apache Mary's Apache group
 GROUP
         598916040194
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.
PROMPT> ec2-describe-group apache
```

```
GROUP 598916040194 apache Mary's Apache group
                                                                 FR<sub>0</sub>M
PERMISSION
              598916040194
                              apache ALLOWS tcp
                                                           80
                                                      80
Mary launches seven instances of her web server AMI as members of the apache group.
PROMPT> ec2run ami-fba54092 -n 7 -g apache
RESERVATION r-0592776c 598916040194 default
INSTANCE i-cfd732a6 ami-fba54092 pending 0 m1.small 2007-07-11T16:4
INSTANCE i-cfd732a7 ami-fba54092 pending 0 ml.small 2007-07-11T16:4
INSTANCE i-cfd732a8 ami-fba54092 pending 0 ml.small 2007-07-11T16:4
INSTANCE i-cfd732a9 ami-fba54092 pending 0 m1.small 2007-07-11T16:4
INSTANCE i-cfd732aa ami-fba54092 pending 0 m1.small 2007-07-11T16:4
INSTANCE i-cfd732ab ami-fba54092 pending 0 m1.small 2007-07-11T16:4
INSTANCE i-cfd732ac ami-fba54092 pending 0 m1.small 2007-07-11T16:4
PROMPT> ec2din i-cfd732a6
RESERVATION
                 r-0592776c
                                   598916040194
INSTANCE
                  i-cfd732a6
                                   ami-fba54092
                                                       ec2-67-202-51-24
m1.small 2007-07-11T16:40:44+0000
Being as paranoid as Albert, Mary uses the Linux and UNIX nmap command to confirm the permissic
s 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
Interesting ports on ec2-67-202-51-245.compute-1.amazonaws.com (67
 (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
Mary verifies her web server can be reached.
$ 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.
Escape character is '^]'.
Mary can reach her web server.
Mary creates a separate group for her application server.
PROMPT> ec2-add-group appserver -d "Mary's app server"
```

GROUP appserver Mary's app server Mary starts twenty instances as members of appserver group. PROMPT> ec2run ami-e3a5408a -n 20 -g appserver Mary grants network access between her web server group and the application server group. PROMPT> ec2-authorize appserver -o apache -u AIDADH4IGTRXXKCD **GROUP** appserver **PERMISSION** appserver ALLOWS all **FROM** USER AIDADH4IGTRXXKC Mary verifies access to her app server is restricted by port scanning one of the application servers usin s 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 All 100 scanned ports on ec2-67-202-51-162.compute-1.amazonaws.com Nmap finished: 1 IP address (1 host up) scanned in 31.008 seconds Mary confirms that her web servers have access to her application servers. A. She (temporarily) grants SSH access from her workstation to the web server group: PROMPT> ec2-authorize apache -P tcp -p 22 -s 126.52.1.130/32 B. She logs in to one of her web servers and connects to an application server on TCP port 8080. \$ 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 Escape character is '^l' C. Satisfied with the setup, she revokes SSH access to the web server group. PROMPT> ec2-revoke apache -P tcp -p 22 -s 126.52.1.130/32 10 Mary repeats these steps to create the database server group and to grant access between the application

Related Topics

• Network Security Concepts

Using Regions and Availability Zones

Topics

•

API Overview

- Describing Regions and Availability Zones
- How to Configure Your Environment
- Launching Instances in Specific Availability Zones
- Related Topics

This section describes how to work with Regions and Availability Zones.



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 <u>Amazon EC2 Product Page</u>.

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 <u>Availability Zones</u>.

Describing Regions and Availability Zones

This section describes how to determine which Regions and Availability Zones are available.

How To

To find Regions

1. If you are using SOAP, construct the following request:

```
<DescribeRegions
xmlns="http://ec2.amazonaws.com/doc/2009-10-31/">
  <regionSet/>
  <DescribeRegions>
```

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=DescribeRegions
&...auth parameters...
```

3. View output similar to the following:

To find Availability Zones

1. If you are using SOAP, construct the following request:

```
<DescribeAvailabilityZones xmlns="http://ec2.amazonaws.com/doc/2009
   <availabilityZoneSet/>
   <DescribeAvailabilityZones>
```

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=DescribeAvailabilityZones
&...auth parameters...
```

```
<DescribeAvailabilityZonesResponse xmlns="http://ec2.amazonaws.com/</pre>
  <availabilityZoneInfo>
    <item>
      <zoneName>us-east-la</zoneName>
      <zoneState>available</zoneState>
    </item>
    <item>
      <zoneName>us-east-1b</zoneName>
      <zoneState>available</zoneState>
    </item>
    <item>
      <zoneName>us-east-1c
      <zoneState>available/zoneState>
    </item>
    <item>
      <zoneName>us-east-1d</zoneName>
      <zoneState>available</zoneState>
    </item>
  </availabilityZoneInfo>
</DescribeAvailabilityZonesResponse>
```

How to Configure Your Environment

After choosing a Region and one or more Availability Zones, you should set up your environment.



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.



Availability Zones are not the same across accounts. The Availability Zone useast-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 <u>Amazon EC2 product page</u>.

How To

To launch an instance in a specific Availability Zone

1. If you are using SOAP, construct the following request:

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
```

```
?Action=RunInstances
&ImageId=ami-id
&MaxCount=1
&MinCount=1
&KeyName=keypair-name
&Placement.AvailabilityZone=zone
&...auth parameters...
```

```
<RunInstancesResponse xmlns="http://ec2.amazonaws.com/doc/2009-10-3</pre>
  <reservationId>r-47a5402e</reservationId>
  <ownerId>AIDADH4IGTRXXKCD</ownerId>
  <qroupSet>
    <item>
      <groupId>default
    </item>
  </groupSet>
  <instancesSet>
    <item>
      <instanceId>i-2ba64342</instanceId>
      <imageId>ami-60a54009</imageId>
      <instanceState>
        <code>0</code>
        <name>pending</name>
      </instanceState>
      <privateDnsName></privateDnsName>
      <dnsName></dnsName>
      <keyName>example-key-name</keyName>
      <amiLaunchIndex>0</amiLaunchIndex>
      <instanceType>m1.small</instanceType>
      <launchTime>2007-08-07T11:51:50.000Z</launchTime>
      <placement>
        <availabilityZone>us-east-1b</availabilityZone>
      </placement>
    </item>
  </instancesSet>
</RunInstancesResponse>
```

Related Topics

- Availability Zones
- Region and Availability Zone FAQ

Using Amazon Elastic Block Store

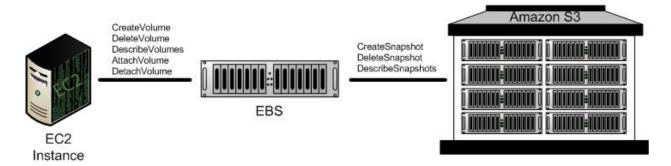
Topics

- Amazon EBS API Overview
- How to Create an Amazon EBS Volume
- How to Attach the Volume to an Instance
- How to Describe Volumes and Instances
- How to Make an Amazon EBS Volume Available for Use
- How to Create an Amazon EBS Snapshot
- How to Describe Snapshots
- How to Modify Snapshot Permissions
- How to Detach an Amazon EBS Volume from an Instance
- How to Delete an Amazon EBS Snapshot
- How to Delete an Amazon EBS Volume
- Related Topics

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.



- **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).
- **ModifySnapshotAttribute**—Modifies permissions for a snapshot. You can specify one or more users or all to make the snapshot public.
- **DescribeSnapshotAttribute**—Describes permissions for a snapshot.

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.



You can also create and attach Amazon EBS volumes when launching instances.

To create an Amazon EBS volume

1. If you are using SOAP, construct the following request:

```
<CreateVolume xmlns="http://ec2.amazonaws.com/doc/2009-10-31/">
    <size>size</size>
    <availabilityZone>zone</availabilityZone>
</CreateVolume>
```

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=CreateVolume
&Size=size
&AvailabilityZone=zone
&...auth parameters...
```

How to Attach the Volume to an Instance

This section describes how to attach a volume that you created to an instance.



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.

To attach an Amazon EBS volume

1. If you are using SOAP, construct the following request:

```
<AttachVolume xmlns="http://ec2.amazonaws.com/doc/2009-10-31/">
    <volumeId>volume-id</volumeId>
        <instanceId>instance-id</instanceId>
        <device>device</device>
</AttachVolume>
```

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=AttachVolume
&VolumeId=volume-id
&InstanceId=instance-id
&Device=device
&...auth parameters...
```

How to Describe Volumes and Instances

After creating Amazon EBS volumes and attaching them to instances, you should verify they are available.

To view information about an Amazon EBS volume

1. If you are using SOAP, construct the following request:

```
<DescribeVolumes xmlns="http://ec2.amazonaws.com/doc/2009-10-31/">
</DescribeVolumes>
```

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=DescribeVolumes
&...auth parameters...
```

```
<DescribeVolumesResponse xmlns="http://ec2.amazonaws.com/doc/2009-1</pre>
<volumeSet>
  <item>
    <volumeId>vol-4282672b/volumeId>
    <size>800</size>
    <status>in-use</status>
    <createTime>2008-05-07T11:51:50.000Z</createTime>
    <attachmentSet>
      <item>
        <volumeId>vol-4282672b/volumeId>
        <instanceId>i-6058a509</instanceId>
        <size>800</size>
        <snapshotId>snap-12345678</snapshotId>
        <availabilityZone>us-east-la</availabilityZone>
        <status>attached</status>
        <attachTime>2008-05-07T12:51:50.000Z</attachTime>
      </item>
    </attachmentSet>
  </item>
</volumeSet>
```

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.



Note

To enable the instance to reconnect to an Amazon EBS volume on reboot, add the device to the fstab or create a script that automatically mounts the volume on boot.

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.



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.

By default, only you can launch volumes from your snapshots. However, you can choose to share your snapshots with specific users or make them public.

For information on creating an Amazon EBS volume from a snapshot, see Amazon Elastic Block Store.

To create a snapshot

1. If you are using SOAP, construct the following request:

```
<CreateSnapshot xmlns="http://ec2.amazonaws.com/doc/2009-10-31/">
    <volumeId>volume-id</volumeId>
    </CreateSnapshot>
```

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=CreateSnapshot
&VolumeId=volume-id
&...auth parameters...
```

How to Describe Snapshots

This section describes how to view snapshots that you created.



When the snapshot is complete, its status will change to completed and the percentage will change to 100%.

To describe snapshots

1. If you are using SOAP, construct the following request:

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=DescribeSnapshots
&SnapshotId=snapshot-id
&...auth parameters...
```

How to Modify Snapshot Permissions

This section describes how to modify permissions for snapshots that you created.



When the snapshot is complete, its status will change to completed and the percentage will change to 100%.

To describe snapshots

1. If you are using SOAP, construct the following request:

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=DescribeSnapshotAttribute
&SnapshotId=snapshot-id
&...auth parameters...
```

3. View output similar to the following:

4. If you are using SOAP, construct the following request to make the snapshot public:

5. If you are using Query, construct the following request to make the snapshot public:

```
https://ec2.amazonaws.com/
?Action=ModifySnapshotAttribute
&SnapshotId.1=snapshot-id
&Add.1=all
&...auth parameters...
```

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.



Note

If an Amazon EBS volume is the root device of an instance, it cannot be detached.

To detach an Amazon EBS volume

1. If you are using SOAP, construct the following request:

```
<DetachVolume xmlns="http://ec2.amazonaws.com/doc/2009-10-31/">
    <volumeId>volume-id</volumeId>
        <instanceId>instance-id</instanceId>
    </DetachVolume>
```

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=DetachVolume
&VolumeId=volume-id
&InstanceId=instance-id
&...auth parameters...
```

3. View output similar to the following:

To verify the volume is no longer attached to the instance, see <u>How to Describe Snapshots</u>.

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.

To delete a snapshot

1. If you are using SOAP, construct the following request:

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=DeleteSnapshot
&SnapshotId=snapshot-id
&...auth parameters...
```

```
<DeleteSnapshotResponse xmlns="http://ec2.amazonaws.com/doc/2009-10
  <return>true</return>
  </DeleteSnapshotResponse>
```

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.

To delete a volume

1. If you are using SOAP, construct the following request:

```
<DeleteVolume xmlns="http://ec2.amazonaws.com/doc/2009-10-31/">
    <volumeId>volume-id

/DeleteVolume>
```

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=DeleteVolume
&VolumeId=volume-id
&...auth parameters...
```

```
<DeleteVolumeResponse xmlns="http://ec2.amazonaws.com/doc/2009-10-3
    <return>true</return>
    </DeleteVolumeResponse>
```

Related Topics

• Amazon Elastic Block Store

Using Auto Scaling, Elastic Load Balancing, and Amazon CloudWatch

Topics

•

Auto Scaling

- Elastic Load Balancing
- Amazon CloudWatch

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.

How To

To launch an instance

1. If you are using SOAP, construct the following request:

```
<RunInstances xmlns="http://ec2.amazonaws.com/doc/2009-10-31/">
    <imageId>ami-id</imageId>
        <minCount>1</minCount>
        <maxCount>1</maxCount>
        <keyName>keypair-name</keyName>
        <groupSet/>
        <monitoring>enabled</monitoring>
</RunInstances>
```

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=RunInstances
&ImageId=ami-id
&MaxCount=1
&MinCount=1
&KeyName=keypair-name
&Monitoring.Enabled
&...auth parameters...
```

3. View output similar to the following:

```
<code>0</code>
        <name>pending</name>
      </instanceState>
      <privateDnsName></privateDnsName>
      <dnsName></dnsName>
      <keyName>example-key-name</keyName>
      <amiLaunchIndex>0</amiLaunchIndex>
      <instanceType>m1.small</instanceType>
      <launchTime>2007-08-07T11:51:50.000Z</launchTime>
      <placement>
        <availabilityZone>us-east-1b</availabilityZone>
      </placement>
      <monitoring>
        <enabled>true</enabled>
      </monitoring>
    </item>
  </instancesSet>
</RunInstancesResponse>
```

Using Public Data Sets

Topics

•

Finding Public Data Sets

- Launching an Instance
- Launching a Public Data Set Volume
- Mounting the Public Data Set Volume
- Related Topics

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 <u>Launching</u> and <u>Using Instances</u>.

Launching a Public Data Set Volume

To use a public data set, you launch an Amazon EBS volume, specifying its snapshot ID.

How To

To create an Amazon EBS volume

1. If you are using SOAP, construct the following request:

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=CreateVolume
&AvailabilityZone=zone
&SnapshotId=public-data-set-snapshot-id
&...auth parameters...
```

3. View output similar to the following:

Mounting the Public Data Set Volume

Mount the volume as you normally do. For more information, see <u>How to Make an Amazon EBS Volume Available for Use</u>.

Related Topics

• Public Data Set Concepts

Reserving Amazon EC2 Instances

Topics

•

How to Find and Purchase Reserved Instances

• Related Topics

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.

How To

To find and purchase a Reserved Instance

1. If you are using SOAP, construct the following request:

```
<DescribeReservedInstancesOfferings xmlns="http://ec2.amazonaws.com</pre>
```

2. If you are using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=DescribeReservedInstancesOfferings
&...auth parameters...
```

3. View output similar to the following:

4. To purchase a Reserved Instance using SOAP, construct a request similar to the following:

5. To purchase a Reserved Instance using Query, construct a request similar to the following:

```
https://ec2.amazonaws.com/
?Action=PurchaseReservedInstancesOffering
&OfferingId.1=offering-12345678&instanceCount.1=2
&...auth parameters...
```

6. View output similar to the following:

```
<PurchaseReservedInstancesOffering xmlns="http://ec2.amazonaws.com/
<reservedInstancesId>reservation-1001001</reservedInstancesId>
</PurchaseReservedInstancesOffering>
```

7. To verify the purchase using SOAP, construct the following request:

```
<DescribeReservedInstances xmlns="http://ec2.amazonaws.com/doc/2009</pre>
```

8. To verify the purchase using Query, construct the following request:

```
https://ec2.amazonaws.com/
?Action=DescribeReservedInstances
&...auth parameters...
```

9. View output similar to the following:

Related Topics

- Reserved Instance Concepts
- Reserved Instances FAQs

Using Amazon Virtual Private Cloud

Topics

•

How To

• Related Topics

Amazon Virtual Private Cloud enables you to designate your own private resources within the AWS cloud, and then connect those resources directly to your own data center using a VPN connection. With regards to Amazon EC2, this mean you can launch instances in your Amazon VPC *virtual private cloud* (VPC), and those instances are isolated from compute capacity being used by other EC2 users. You can also separate the instances into different subnets to provide network isolation of the instances within your private VPC.

This section describes how to launch instances into a VPC that you've created. For detailed information about Amazon VPC and creating a VPC and subnets, go to the <u>Amazon Virtual Private Cloud Getting Started Guide</u>.

How To

To launch an instance in a VPC

1. If you are using SOAP, construct the following request, with the ID of the Amazon VPC subnet where you want the instance launched:

```
<RunInstances xmlns="http://ec2.amazonaws.com/doc/2009-10-31/">
        <imageId>ami-id</imageId>
        <minCount>1</minCount>
        <maxCount>1</maxCount>
        <keyName>keypair-name</keyName>
        <groupSet/>
        <monitoring/>
        <subnetId>subnet-id</subnetId>
</RunInstances>
```

2. If you are using Query, construct the following request, with the ID of the Amazon VPC subnet where you want the instance launched:

```
https://ec2.amazonaws.com/
?Action=RunInstances
&ImageId=ami-id
&MaxCount=1
&MinCount=1
&KeyName=keypair-name
&SubnetId=subnet-id
&...auth parameters...
```

3. View output similar to the following:

```
<name>pending</name>
      </instanceState>
      <privateDnsName></privateDnsName>
      <dnsName></dnsName>
      <keyName>example-key-name</keyName>
      <amiLaunchIndex>0</amiLaunchIndex>
      <instanceType>m1.small</instanceType>
      <launchTime>2009-08-07T11:51:50.000Z</launchTime>
      <placement>
        <availabilityZone>us-east-1b</availabilityZone>
      </placement>
      <monitoring>
        <enabled>false/enabled>
      </monitoring>
      <subnetId>subnet-la2b3d4d</subnetId>
      <vpcId>vpc-6e7f8a92</pcId>
      <privateIpAddress>10.0.0.1</privateIpAddress>
      <ipAddress></ipAddress>
    </item>
  </instancesSet>
</RunInstancesResponse>
```

Each instance you launch in a VPC is assigned a private IP address within the range of IP addresses you specified when you created the subnet. The instance has no public IP address or DNS names. For information about programmatically creating a VPC and subnets, go to the <u>Amazon Virtual Private Cloud Developer Guide</u>.



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.

Related Topics

• Amazon Virtual Private Cloud

Amazon EC2 Programming

Topics

•

Using the SOAP API

- <u>Using the Query API</u>
- Available Libraries

This section provides an overview of the SOAP and Query APIs.

Using the SOAP API

Topics

•

WSDL and Schema Definitions

- Programming Language Support in Amazon EC2
- Request Authentication
- The Response Structure

WSDL and Schema Definitions

The Amazon EC2 web service can be accessed using the SOAP web services messaging protocol. This interface is described by a Web Services Description Language (WSDL) document which defines the operations and security model for the service. The WSDL references an XML Schema document which strictly defines the data types that might appear in SOAP requests and responses.

All schemas have a version number (the latest is 2009-10-31). The version number appears in the URL of a schema file, and in a schema's target namespace. This makes upgrading easy by differentiating requests based on the version number.



In addition to the latest version, the service will support the older versions for some time, allowing customers plenty of time to upgrade.

The Amazon EC2 services API WSDL is available from the web at 'http://ec2.amazonaws.com/doc/<version>/ec2.wsdl' where *version* is the version of the API. At the time this document was released, the current API version was 2009-10-31, which is available at http://ec2.amazonaws.com/doc/2009-10-31/AmazonEC2.wsdl

The following are additional web service references.

- Web Service Description Language (WSDL)
- WS-Security BinarySecurityToken Profile

Programming Language Support in Amazon EC2

Since the SOAP requests and responses in the Amazon EC2 Web Service follow current standards, any programming language with the appropriate library support can be used. Languages known to have this support include C++, C#, Java, Perl, Python and Ruby.

Request Authentication

To prevent in-flight tampering, all SOAP requests should be sent over HTTPS. In addition, the service complies with the current WS-Security standard, requiring SOAP request messages to be hashed and signed for integrity and non-repudiation. WS-Security defines profiles which are used to implement various levels of security. Amazon EC2 secure SOAP messages use the BinarySecurityToken profile, consisting of an X.509 certificate with an RSA public key.

The following is the content of an insecure RunInstances operation:

To secure the request, we add the BinarySecurityToken element. The Java libraries we supply rely on the Apache Axis project for XML security, canonicalization, and SOAP support. The Sun Java Web Service Developer's Pack supplies libraries of equivalent functionality.

The secure version of the request begins with the following:

```
<ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/]</pre>
        <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmlds</pre>
        <ds:Reference URI="#id-17984263">
          <ds:Transforms>
             <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc</pre>
          </ds:Transforms>
          <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmlds:</pre>
          <ds:DigestValue>0pjZ1+TvgPf6uG7o+Yp3l2YdGZ4=</ds:DigestValue</pre>
        </ds:Reference>
        <ds:Reference URI="#id-15778003">
          <ds:Transforms>
             <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc</pre>
          </ds:Transforms>
          <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmlds:</pre>
          <ds:DigestValue>HhRbxBBmc200348f8nLNZyo4A0M=</ds:DigestValue</pre>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>bmVx24Qom4kd9QQtclxWIlgLk4QsQBPaKESi79x479xgk
      jjHKZKEQRCOlLVyODn5ZL1RlMHsv+OzJzzvIJFTq3LQKNrzJzsNe</ds:Signati
      <ds:KeyInfo Id="KeyId-17007273">
        <wsse:SecurityTokenReference</pre>
             xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-20
          <wsse:Reference URI="#CertId-1064304"</pre>
                           ValueType="http://docs.oasis-open.org/wss/20
          </wsse:Reference>
        </wsse:SecurityTokenReference>
      </ds:KeyInfo>
    </ds:Signature>
    <wsu:Timestamp</pre>
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401
      <wsu:Created>2006-06-09T10:57:35Z</wsu:Created>
      <wsu:Expires>2006-06-09T11:02:35Z</wsu:Expires>
    </wsu:Timestamp>
  </wsse:Security>
</SOAP-ENV:Header>
```

If you are matching this against requests generated by Amazon EC2 supplied libraries, or those of another vendor, the following are the most important elements:

Elements

• **BinarySecurityToken**—Contains the X.509 certificate in base64 encoded PEM format

- **Signature**—Contains an XML digital signature created using the canonicalization, signature algorithm, and digest method
- **Timestamp**—Requests to Amazon EC2 are valid within 5 minutes of this value to help prevent replay attacks

The Response Structure

In response to a request, the Amazon EC2 web service returns an XML data structure that conforms to an XML schema defined as part of the Amazon EC2 WSDL. The structure of a XML response is specific to the associated request. In general, the response data types are named according to the operation performed and whether the data type is a container (can have children). Examples of containers include groupSet for security groups and instancesSet for instances. Item elements are children of containers and their contents vary according to the container's role.

The following is an example response:

```
<RunInstancesResponse xmlns="http://ec2.amazonaws.com/doc/2009-10-31/">
  <reservationId>r-47a5402e</reservationId>
  <ownerId>UYY3TLBUXIEON5NQVUUX60MPWBZIQNFM</ownerId>
  <groupSet>
    <item>
      <groupId>default
    </item>
 </groupSet>
 <instancesSet>
    <item>
      <instanceId>i-2ba64342</instanceId>
      <imageId>ami-60a54009</imageId>
      <instanceState>
        <code>0</code>
        <name>pending</name>
     </instanceState>
      <dnsName></dnsName>
    </item>
    <item>
      <instanceId>i-2bc64242</instanceId>
      <imageId>ami-60a54009</imageId>
      <instanceState>
        <code>0</code>
        <name>pending</name>
      </instanceState>
      <dnsName>ec2-67-202-51-176.compute-1.amazonaws.com </dnsName>
    </item>
    <item>
      <instanceId>i-2be64332</instanceId>
      <imageId>ami-60a54009/imageId>
      <instanceState>
```

Using the Query API

Topics

•

Query Parameters

• Query API Authentication

Query requests are HTTP or HTTPS requests that use the HTTP verb GET or POST and a Query parameter named Action or Operation. Action is used throughout this documentation, although Operation is supported for backward compatibility with other AWS Query APIs.

Query Parameters

Each Query request must include some common parameters to handle authentication and selection of an action. For more information, go to the <u>Amazon Elastic Compute Cloud API Reference</u>.

Some operations take lists of parameters. These lists are specified using the param.n notation. Values of n are integers starting from 1.

Query API Authentication

You can send Query requests over either HTTP or HTTPS. Regardless of which protocol you use, you must include a signature in every Query request. This section describes how to create the signature. The method described in the following procedure is known as *signature version 2*.



TOTAL Caution

If you are currently using signature version 1: Version 1 is deprecated, and you should move to signature version 2 as soon as possible. For information about the deprecation schedule and the differences between signature version 2 and version 1, go to Making Secure Requests to Amazon Web Services.

To create the signature

- 1. Create the canonicalized query string that you need later in this procedure:
 - a. Sort the UTF-8 query string components by parameter name with natural byte ordering.
 - The parameters can come from the GET URI or from the POST body (when Content-Type is application/x-www-form-urlencoded).
 - b. URL encode the parameter name and values according to the following rules:
 - Do not URL encode any of the unreserved characters that RFC 3986 defines.
 - These unreserved characters are A-Z, a-z, 0-9, hyphen (), underscore ($_$), period (.), and tilde (\sim).
 - Percent encode all other characters with %XY, where X and Y are hex characters 0-9 and uppercase A-F.
 - Percent encode extended UTF-8 characters in the form %XY%ZA....

■ Percent encode the space character as %20 (and not +, as common encoding schemes do).



Note

Currently all AWS service parameter names use unreserved characters, so you don't need to encode them. However, you might want to include code to handle parameter names that use reserved characters, for possible future use.

- c. Separate the encoded parameter names from their encoded values with the equals sign (=) (ASCII character 61), even if the parameter value is empty.
- d. Separate the name-value pairs with an ampersand (&) (ASCII code 38).
- 2. Create the string to sign according to the following pseudo-grammar (the "\n" represents an ASCII newline).

The HTTPRequestURI component is the HTTP absolute path component of the URI up to, but not including, the query string. If the HTTPRequestURI is empty, use a forward slash (/).

3. Calculate an RFC 2104-compliant HMAC with the string you just created, your Secret Access Key as the key, and SHA256 or SHA1 as the hash algorithm.

For more information, go to http://www.rfc.net/rfc2104.html.

- 4. Convert the resulting value to base64.
- 5. Use the resulting value as the value of the <code>Signature</code> request parameter.



The final signature you send in the request must be URL encoded as specified in RFC 3986 (for more information, go to http://www.rfc.net/rfc3986.html). If your toolkit URL encodes your final request, then it handles the required URL encoding of the signature. If your toolkit doesn't URL encode the final request, then make sure to URL encode the signature before you include it in the request. Most importantly, make sure the signature is URL encoded *only once*. A common mistake is to URL encode it manually during signature formation, and then again when the toolkit URL encodes the entire request.

Example DescribeImages Request

```
https://ec2.amazonaws.com/
?Action=DescribeImages
&ImageId.1=ami-2bb65342
&Version=2009-10-31
&Expires=2008-02-10T12%3A00%3A00Z
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&AWSAccessKeyId=<Your AWS Access Key ID>
```

Following is the string to sign.

```
GET\n
ec2.amazonaws.com\n
/\n
AWSAccessKeyId=<Your AWS Access Key ID>
&Action=DescribeImages
&Expires=2008-02-10T12%3A00%3A00Z
&ImageId.1=ami-2bb65342
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Version=2009-10-31
```

Following is the signed request.

```
https://ec2.amazonaws.com/
?Action=DescribeImages
&ImageId.1=ami-2bb65342
&Version=2009-10-31
&Expires=2008-02-10T12%3A00%3A00Z
&Signature=<URLEncode(Base64Encode(Signature))>
&SignatureVersion=2
&SignatureMethod=HmacSHA256
```

Available Libraries

AWS provides libraries, sample code, tutorials, and other resources for software developers who prefer to build applications using language-specific APIs instead of SOAP and Query. These libraries provide basic functions (not included in the APIs), such as request authentication, request retries, and error handling so that it is easier to get started. Libraries and resources are available for the following languages:

- Java
- <u>PHP</u>
- Ruby
- Windows and .NET

For libraries and sample code in all languages, go to **Sample Code & Libraries**.

Technical FAQ

Topics

•

General Information FAQ

- Instances and AMIs Information FAQ
- Operation Information FAQ
- Instance Types and Architectures FAQ
- IP Information FAQ
- Region and Availability Zone FAQ
- Windows Instances FAQ
- Monitoring, Errors, and Unexpected Behavior FAQ
- Reserved Instances FAQs
- Paid AMIs FAQ
- Kernels, RAM Disks, and Block Device Mappings FAQ
- Error Messages FAQ
- Miscellaneous FAQ

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 <u>Amazon EC2 Instance</u> Request Form and your request will be considered.

How do I sign a request?

Information on signing SOAP requests is provided in <u>Request Authentication</u>. Information on signing Query requests is provided in <u>Query API Authentication</u>

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 <u>Amazon Elastic Compute Cloud Getting Started Guide</u>.

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.

If you need faster launch times, consider using AMIs backed by Amazon EBS.

Can I get a bigger/smaller/differently optimized virtual machine?

Yes. For more information, see <u>Instance Types</u>.

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.

Instances and AMIs Information FAQ

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 fail?

The instances terminate and need to be relaunched. The data on the instance stores is lost. Any data on Amazon EBS volumes (including the root device for Amazon EBS-backed instances) is preserved.

Always replicate important data, use Amazon EBS, or store it in Amazon S3.

What are the differences between stopping and terminating an instance?

The instance will first perform a normal shutdown in both cases. If the instance was stopped, it will then transition to a stopped state, all of its Amazon EBS volumes will remain attached, and it can then be started again at a later time. If the instance was terminated, attached Amazon EBS volumes will be deleted where the associated deleteOnTerminate instance attribute is set to true. The instance itself will also be deleted, and the instance cannot be started again at a later date.

If you wish to disable termination via the TerminateInstances API for increased safety, ensure that the disableApiTermination attribute is set to true for the instance. To control the behavior of an on instance shutdown such as shutdown – h in linux or shutdown in windows set the instanceInitiatedShutdownBehavior instance attribute to stop or terminate as desired. Instances that have Amazon EBS volume root devices will default to stop, instances with instance-store root devices will always be terminated as the result of an on instance shutdown.

How does stopping and starting an instance effect how it is charged?

You will not be charged for additional instance hours while the instance is in a stopped state. A full instance hour will be charged for every transition from a stopped state to a running state, even if this happens multiple times within a single hour. If the instance type was changed while the instance was stopped, you will be charged at the new instance type's rate once the instance has been started. All of the associated Amazon EBS usage of your instance, including root device usage, will be charged at the typical Amazon EBS prices.

What can I do with a stopped instance?

Amazon EBS volumes can be attached to or detached from the instance. The CreateImage call can be used to create an AMI from the instance. The kernel, ramdisk, and instance type can be changed with the ModifyInstanceAttribute call. The instance can be started with the StartInstances call and can be terminated with the TerminateInstances call.

What limitations exist for stopped instances?

In addition to the limit on running instances, there is an additional limit on the overall number of instances that you can have (whether running, stopped, or in any other state except for terminated.) This overall instance limit is 2 times your running instance limit. The running instance limit can be increased upon request via the instance limit request form.

Do I need to share the Amazon EBS snapshots that an AMI references in order to share the AMI?

No, only the AMI itself needs to be shared. The system will automatically provide the instance access to the referenced Amazon EBS snapshots for the purposes of the launch.

How can I control how block devices are exposed to my instance?

You may control how block devices are exposed in your instance by configuring the block device mapping in a RunInstances call, in a stopped instance by using the ModifyInstanceAttribute call, and in an AMI by having a snapshot attached during a CreateImage call. Using these calls, you can configure the your instance to start with a specific snapshot mounted, use the rootDeviceName field to

specify the name of the instance's the root device when the instance is booted, and you may also gain access to the local instance stores as instance stores are never exposed by default on instances backed by Amazon EBS. An example of an instance block device mapping would be device /dev/sdc mapped to vol-1234abcd.

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://169.254.169.254/2009-10-31/ to retrieve various instance-specific metadata, including the instance ID. For more information, see Instance Instance Insta

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.

Is there a way to run a script on instance termination?

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.

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:

Properties for configuring a proxy are described in the following table.

Setting	Description	
https.proxyHost	HTTPS proxy host	
https.proxyPort	HTTPS proxy port	
http.proxyHost	HTTP proxy host	
http.proxyPort	HTTP proxy port	
http.proxyRealm	Proxy realm (https and http)	
http.proxyUser	Proxy username (https and http)	
http.proxyPass	Proxy password (https and http)	



Note

 ${\tt https.proxyHost}\ should\ be\ used\ when\ {\tt EC2_URL}\ points\ to\ an\ https.host,\ and\ http.proxyHost\ when\ {\tt EC2_URL}\ points\ to\ an\ http\ host.$

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 three families: standard, High-CPU, and High-Memory. For more information, refer to <u>Instance Types</u>.



Tip

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. A 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 instance stores on large and extra large instances have higher and more consistent I/O performance than the original (small) instance. Amazon EBS volumes perform consistently for all instance types.



The first write to any given block of the disk will be slower than subsequent writes. For more information, see <u>Disk Performance Optimization</u>

Can I RAID the spindles exposed on large and extra large instances?

Yes, you can use software RAID on top of the exposed spindles.



The initial RAID setup might take a long time. For more information, see <u>Disk</u> <u>Performance Optimization</u>

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 <u>DynDNS</u> or <u>ZoneEdit</u>. 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://169.254.169.254/2009-10-31//meta-data/local-ipv4

To obtain the public IP address:

curl http://169.254.169.254/2009-10-31//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 <u>Amazon EC2 Elastic IP Address Request Form</u>. 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:

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 the EU Region, 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 zateam@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.



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.



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.

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 EU 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 <u>Resources</u>.

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.

For more advanced monitoring, consider Amazon CloudWatch. Amazon CloudWatch runs a monitoring services that collects raw measurement data, such as CPUUtilization (percentage of Amazon EC2 compute units used by an instance) and DiskWriteBytes (number of bytes written in a minute). For more information, refer to the *Amazon CloudWatch Developer Guide*.

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

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.



The ec2-describe-instances command line tool does not provide this information. Use the $\neg v$ 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 <u>AWS forums</u>.

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

Do you support paid AMIs that are backed by Amazon EBS?

Not at this time.



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.



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.

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://169.254.169.254/2009-10-31/?

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://169.254.169.254/2009-10-31/)
- 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



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 <u>NTP</u>.

Miscellaneous FAQ

What runlevel do instances start in?

All Linux instances are started in runlevel 4, regardless of the instance configuration.

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

- Metadata Categories Windows Configuration Service

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 <u>reservation</u> (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 .	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
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	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.

Resources

The following table describes which Amazon EC2 resources are global, Regional, or Availability Zone-based.

Resource	Туре	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.	
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.	

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.

.



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 <u>Instance Storage</u>.
- **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.

```
<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.

Amazon EBS-backed AMI

An instance launched from an AMI backed by Amazon EBS uses an Amazon EBS volume as its root device. See *Amazon EBS*.

Amazon S3-backed AMI

An instance launched from an Amazon S3backed AMI uses an instance store as its root device. See *instance store*.

Availability Zone

A distinct location within a Region that is engineered to be insulated from failures in other Availability Zones and provides inexpensive, low latency network connectivity to other Availability Zones in the same Region.

compute unit

An Amazon-generated measure that enables you to evaluate the CPU capacity of different Amazon EC2 instance types.

EBS

See Amazon EBS.

Elastic Block Store

See *Amazon EBS*.

elastic IP address

A static public IP address designed for dynamic cloud computing. Elastic IP addresses are associated with your account, not specific instances. Any elastic IP addresses that you associate with your account remain associated with your account until you explicitly release them. Unlike traditional static IP addresses, however, elastic IP addresses allow you to mask instance or Availability Zone failures by rapidly remapping your public IP addresses to any instance in your account.

ephemeral store

See instance store.

explicit launch permission

Launch permission granted to a specific user.

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.

group

See *security group*.

image

See *Amazon machine image*.

instance

Once an AMI has been launched, the resulting running system is referred to as an instance. All instances based on the same AMI start out identical and any information on them is lost when the instances are terminated or fail.

instance store

Every instance includes a fixed amount of storage space on which you can store data. This is not designed to be a permanent storage solution. If you need a permanent storage system, use Amazon EBS.

instance type

A specification that defines the memory, CPU, storage capacity, and hourly cost for an instance. Some instance types are designed for standard applications while others are designed for CPU-intensive applications.

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 *Amazon DevPay Developer Guide*.

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 further discussion elsewhere.		
	You can use this resource regularly.		
Code in text	Inline code samples (including XML) and commands are identified with a special font.		
	You can use the command java -version.		
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/inde. # 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	References to a section in the same document are marked.		
cross references	See <u>Document Conventions</u> .		
Logical	A special font is used for expressions that are important to identify, but are not code.		
values, constants,	If the value is null, the returned response will be false.		
and regular expressions, abstracta			
Product and	Named AWS products and features are identified on first use.		
feature names	Create an Amazon Machine Image (AMI).		
Operations	In-text references to operations.		
	Use the GetHITResponse operation.		
Parameters	In-text references to parameters.		
	The operation accepts the parameter <i>Account ID</i> .		

Response elements	In-text references to responses.		
	A container for one CollectionParent and one or more CollectionItems.		
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 c sample to a command line.		
	% ec2-register < <i>your-s3-bucket</i> >/image.manifest		
	See also <u>Symbol Conventions</u> .		

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. % data = hdfread (start stride edge)
Optional parameters XML variable text	[square brackets]	Within a code description, square brackets denote completely optional commands or parameters. % sed [-n, -quiet] Use square brackets in XML examples to differentiate them from tags. <pre></pre> <
Variables	<arrow brackets=""></arrow>	Within a code sample, arrow brackets denote a variable that must be replaced with a valid value. % ec2-register <your-s3-bucket>/image.manifest</your-s3-bucket>

Index

Α

```
accessing instances, Accessing Instances
account ID
    getting, Using Credentials
addressing, Instance Addressing Concepts, Using Instance Addressing
Amazon CloudWatch
    concepts, Amazon CloudWatch
    using, Using Auto Scaling, Elastic Load Balancing, and Amazon
    CloudWatch
Amazon DevPay, Creating Paid AMIs
Amazon EBS
    concepts, Amazon Elastic Block Store
AMIs
    bundling, Creating an Amazon S3-Based AMI
    creating, Preparing an AMI, Creating an Amazon S3-Based AMI
    creating Amazon EBS-backed, Creating an Amazon EBS-Backed AMI
    information, <u>Instances and AMIs Information FAQ</u>
    paid, Paying for AMIs
    shared, Using Shared AMIs
         finding, How to Find Shared AMIs
         security, Safe Use of Shared AMIs
    sharing, How to Share AMIs
AMIs backed by Amazon EBS
    creating, Creating an Amazon EBS-Backed AMI
API
    Query, <u>Using the Query API</u>
    SOAP, Using the SOAP API
APIs, using, Amazon EC2 Programming
appendix, Appendix
audience, Who Should Read This Guide
authentication
    Query, Query API Authentication
    signature version 2, Query API Authentication
    SOAP, Request Authentication
```

Auto Scaling concepts, Auto Scaling using, Using Auto Scaling, Elastic Load Balancing, and Amazon **CloudWatch** Availability Zones, Region and Availability Zone Concepts, Using Regions and Availability Zones, Region and Availability Zone FAQ B batch processing, Introduction to Amazon Elastic Compute Cloud best practices, Instance Usage block device mapping, Kernels, RAM Disks, and Block Device Mappings FAQ bundling AMIs, Creating an Amazon S3-Based AMI \mathbf{C} categories, Metadata Categories Census data, Public Data Set Concepts certificates creating, **Using Credentials** changes to Amazon EC2, What's New CloudWatch concepts, Amazon CloudWatch using, Using Auto Scaling, Elastic Load Balancing, and Amazon **CloudWatch** computation building block, **Instance Usage** compute resources, measuring, Measuring Compute Resources concepts, Amazon EC2 Concepts Amazon EBS, Amazon Elastic Block Store block storage, Amazon Elastic Block Store public data sets, **Public Data Set Concepts** Reserved Instances, Reserved Instance Concepts Virtual Private Cloud, Amazon Virtual Private Cloud Windows, Windows Instance Types console output, Getting Console Output and Rebooting Instances CPU, I/O Resources creating AMIs, Preparing an AMI, Creating an Amazon S3-Based AMI

creating AMIs backed by Amazon EBS, Creating an Amazon EBS-Backed AMI

```
creating paid AMIs, Creating Paid AMIs
credentials
    using, Using Credentials
D
data retrieval, Data Retrieval
data sets, Public Data Set Concepts, Using Public Data Sets
device mapping, Kernels, RAM Disks, and Block Device Mappings FAQ
DevPay, Creating Paid AMIs
disk
    performance, Disk Performance Optimization
    RAID, RAID Configuration
DNS, internal, Internal DNS Name
\mathbf{E}
elastic IP addresses, <u>Instance Addressing Concepts</u>, <u>Using Instance Addressing</u>
Elastic Load Balancing
    concepts, Elastic Load Balancing
    using, Using Auto Scaling, Elastic Load Balancing, and Amazon
    CloudWatch
errors, Monitoring, Errors, and Unexpected Behavior FAQ
    messages, Error Messages FAQ
F
FAQs, Technical FAQ
     AMIs, Instances and AMIs Information FAQ
    Availability Zones, Region and Availability Zone FAQ
    block device mapping, Kernels, RAM Disks, and Block Device Mappings
    FAO
    errors, Monitoring, Errors, and Unexpected Behavior FAQ
    general, General Information FAQ
    instance types, <u>Instance Types and Architectures FAQ</u>
    instances, Instances and AMIs Information FAO
    IP addresses, IP Information FAQ
    kernels, Kernels, RAM Disks, and Block Device Mappings FAQ
```

```
miscellaneous, Miscellaneous FAQ
    monitoring, Monitoring, Errors, and Unexpected Behavior FAQ
    operations, Operation Information FAQ
    paid AMIs, Paid AMIs FAQ
    proximity, Region and Availability Zone FAQ
    RAM disk, Kernels, RAM Disks, and Block Device Mappings FAQ
    Reserved Instances, Reserved Instances FAQs
    unexpected behaviors, Monitoring, Errors, and Unexpected Behavior FAQ
    Windows, Windows Instances FAQ
fault tolerance, Failure Resilient Application Concepts
    using, Using Auto Scaling, Elastic Load Balancing, and Amazon
    CloudWatch
G
general information, General Information FAQ
glossary, Glossary
H
Human Genome Project data, Public Data Set Concepts
Ι
I/O resources, I/O Resources
instance store, Instance Storage, Instance Storage Usage
instance types, Instance Types and Architectures FAQ
instances
    accessing, Accessing Instances
    addressing, Instance Addressing Concepts, Using Instance Addressing
    information, Instances and AMIs Information FAQ
    launching, Launching and Using Instances
    metadata, Instance Metadata
    rebooting, Getting Console Output and Rebooting Instances
    security, Instance Addressing Concepts, Using Instance Addressing
    sizes, <u>Instance Types</u>
    storage, Instance Storage, Instance Storage Usage
    types, Instance Types
```

```
usage, <u>Instance Usage</u>
introduction, <u>Introduction to Amazon Elastic Compute Cloud</u>
IP address information, <u>IP Information FAQ</u>
```

K

kernels, Kernels, RAM Disks, and Block Device Mappings FAQ

L

\mathbf{M}

```
mapping, block device, Kernels, RAM Disks, and Block Device Mappings FAQ memory, I/O Resources
metadata, Instance Metadata
    categories, Metadata Categories
    retrieval, Metadata Retrieval
miscellaneous FAQs, Miscellaneous FAQ
monitoring, Failure Resilient Application Concepts
    using, Using Auto Scaling, Elastic Load Balancing, and Amazon
    CloudWatch
monitoring information, Monitoring, Errors, and Unexpected Behavior FAQ
```

Ν

NAT, Instance Addressing Concepts, Using Instance Addressing

```
network
    private, Amazon Virtual Private Cloud, Using Amazon Virtual Private
    Cloud
network security, Network Security Concepts, Using Network Security
new features, What's New
0
operations
    information, Operation Information FAQ
output, console, Getting Console Output and Rebooting Instances
overview, Introduction to Amazon Elastic Compute Cloud
P
paid AMIs
    creating, Creating Paid AMIs
    information, Paid AMIs FAQ
Paid AMIs, Paying for AMIs
password
    using, Using Credentials
performance, optimization, Disk Performance Optimization
permissions, Instance Usage
private addresses, Instance Addressing Concepts, Using Instance Addressing
private cloud, Amazon Virtual Private Cloud, Using Amazon Virtual Private
private network, Amazon Virtual Private Cloud, Using Amazon Virtual Private
Cloud
programming language support, Programming Language Support in Amazon
proximity, Region and Availability Zone Concepts, Using Regions and
Availability Zones, Region and Availability Zone FAQ
public addresses, Instance Addressing Concepts, Using Instance Addressing
public data sets
    concepts, Public Data Set Concepts
    using, Using Public Data Sets
```

```
Query
    API, <u>Using the Query API</u>
    authentication, Query API Authentication
    parameters, Query Parameters
R
RAID, RAID Configuration
RAM disk, Kernels, RAM Disks, and Block Device Mappings FAQ
reboot, Getting Console Output and Rebooting Instances
remote access, Accessing Instances
Remote Desktop, Accessing Instances
required knowledge, Required Knowledge and Skills
Reserved Instances
    concepts, Reserved Instance Concepts
    information, Reserved Instances FAOs
    using, Reserving Amazon EC2 Instances
resources, Amazon EC2 Resources
    I/O, I/O Resources
    measuring, Measuring Compute Resources
response structure, The Response Structure
response structure, SOAP, The Response Structure
retrieving metadata, Metadata Retrieval
retrieving user data, User Data Retrieval
S
scalability, Failure Resilient Application Concepts
    using, Using Auto Scaling, Elastic Load Balancing, and Amazon
    CloudWatch
scalable applications, <u>Introduction to Amazon Elastic Compute Cloud</u>
security, Network Security Concepts, Using Network Security
service overview, Introduction to Amazon Elastic Compute Cloud
shared AMIs, Using Shared AMIs
    finding, How to Find Shared AMIs
    security, Safe Use of Shared AMIs
sharing AMIs, How to Share AMIs
```

signature version 2, Query API Authentication

```
sizes of instances, Instance Types
SOAP
    API, Using the SOAP API
    authentication, Request Authentication
     response structure, The Response Structure
     WSDL, WSDL and Schema Definitions
SQL, Windows Instances FAQ
SSH, Accessing Instances
static IPs, Instance Addressing Concepts, Using Instance Addressing
storage, <u>Instance Storage</u>, <u>I/O Resources</u>, <u>Instance Storage Usage</u>
    locations, Storage Locations
suggestions, Instance Usage
\mathbf{T}
temporary events, Introduction to Amazon Elastic Compute Cloud
types of instances, Instance Types
IJ
unexpected behavior information, Monitoring, Errors, and Unexpected Behavior
FAQ
US Census data, <u>Public Data Set Concepts</u>
user data, retrieval, User Data Retrieval
using, Using Amazon EC2
    credentials, Using Credentials
     public data sets, Using Public Data Sets
    Reserved Instances, Reserving Amazon EC2 Instances
     Virtual Private Cloud, Using Amazon Virtual Private Cloud
V
```

Virtual Private Cloud
concepts, <u>Amazon Virtual Private Cloud</u>
using, <u>Using Amazon Virtual Private Cloud</u>
virtual private network, <u>Amazon Virtual Private Cloud</u>, <u>Using Amazon Virtual Private Cloud</u>
<u>Private Cloud</u>
VPN, Amazon Virtual Private Cloud, Using Amazon Virtual Private Cloud

\mathbf{W}

Wikipedia data, <u>Public Data Set Concepts</u>
Windows, <u>Windows Instances FAQ</u>
concepts, <u>Windows Instance Types</u>
Windows AMIs, <u>Windows AMIs</u>
WSDL, <u>WSDL and Schema Definitions</u>

Z

zones, availability, Region and Availability Zone FAQ