
Amazon Elastic Compute Cloud

User Guide

API Version 2013-02-01



Amazon Web Services

Amazon Elastic Compute Cloud: User Guide

Amazon Web Services

Copyright © 2013 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

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

What is Amazon EC2?	1
Getting Started	8
Amazon Machine Images	18
AMI Basics	18
Finding a Suitable AMI	21
Shared AMIs	24
Paid AMIs	26
Creating Amazon EBS-Backed AMIs Using the Console	28
Creating Your Own AMIs	30
Creating Amazon EBS-Backed Linux AMIs	31
Creating Instance Store-Backed Linux/UNIX AMIs	34
Tools You Need	35
From an Existing AMI	36
From a Loopback	40
Creating and Launching an AMI from a Snapshot	48
Creating Paid AMIs	48
Sharing AMIs Safely	58
Copying AMIs	67
Amazon Linux AMIs	71
Enabling Your Own Linux Kernels	77
Instances	86
Instance Basics	87
Families and Types	87
Micro Instances	94
High I/O Instances	101
High Storage Instances	102
Cluster Instances	104
Regions and Availability Zones	116
Supported Platforms	120
Root Device Volume	123
Best Practices	129
Stopping, Starting, and Terminating Instances	130
Spot Instances	132
Getting Started with Spot Instances	133
Viewing Spot Price History	136
Creating a Spot Instance Request	139
Finding Running Spot Instances	143
Canceling Spot Instance Requests	147
Fundamentals of Spot Instances	150
Placing Spot Requests	150
Spot Instance Limits	151
Customizing Your Spot Requests	152
Tracking Spot Requests with Bid Status Codes	153
Tagging Spot Instance Requests	159
Protecting Your Spot Instance Data Against Interruptions	160
Planning for Interruptions	160
Persisting Your Root EBS Partition	160
Walkthroughs: Using Spot Instances with AWS Services	161
Managing Spot Instances with Auto Scaling	161
Tools for Managing Auto Scaling with Spot Instances	162
Launching Spot Instances with Auto Scaling	164
Obtaining Information About the Instances Launched by Auto Scaling	167
Updating the Bid Price for the Spot Instances	171
Scheduling Spot Bid Requests	173
Using Auto Scaling to Get Notifications for Spot Instances	174
Using CloudFormation Templates to Launch Spot Instances	176
Launching Amazon Elastic MapReduce Job Flows with Spot Instances	177
Launching Spot Instances in Amazon Virtual Private Cloud	178

Advanced Tasks	180
Subscribe to Your Spot Instance Data Feed	180
Programming Spot with AWS Java SDK	183
Tutorial: Amazon EC2 Spot Instances	184
Tutorial: Advanced Amazon EC2 Spot Request Management	194
Starting Clusters on Spot Instances	210
Reserved Instances	212
Getting Started with Reserved Instances	213
Tools for Working with Reserved Instances	215
Choosing Reserved Instances Based on Your Usage Plans	217
Buying Reserved Instances	219
Becoming a Buyer	220
Purchasing Reserved Instances	221
Reading Your Statement (Invoice)	227
Understanding Discount Pricing Tiers	229
Obtaining Information About Your Reserved Instances	237
Understanding the Pricing Benefit of Reserved Instances	241
Reserved Instances and Consolidated Billing	242
Reserved Instance Marketplace	243
Selling in the Reserved Instance Marketplace	245
Registering as a Seller	246
Selling Your Reserved Instances	250
After Your Reserved Instance Is Sold	270
Requirements Checklist for Reserved Instances	272
Launching Instances	274
Getting a Key Pair	274
Have AWS Create the Key Pair	275
Generate Your Own Key	279
Verifying Your Key Pair	283
Launching an Instance from an AMI	284
Creating and Launching an AMI from a Snapshot	290
Connecting to Amazon EC2 Instances	293
Authorizing Network Access to Your Instances	293
Connecting to Your Linux/UNIX Instances Using SSH	295
Connecting to Linux/UNIX Instances from Windows Using PuTTY	301
Connecting to Windows Instances	306
Importing and Exporting Instances	308
Importing EC2 Instances	308
Before You Get Started	309
Using the Amazon EC2 VM Import Connector to Import Your Virtual Machine to Amazon EC2	310
Using the Command Line Tools to Import Your Virtual Machine to Amazon EC2	324
Troubleshooting Instance Importation	336
Exporting EC2 Instances	338
Managing Instances	340
Auto Scaling and Load Balancing Your Instances	340
Monitoring Your Instances	342
Monitoring Your Instances with CloudWatch	343
Monitoring the Status of Your Instances	354
Monitoring Instances with Status Checks	354
Monitoring Events for Your Instances	358
Ensuring Idempotency	363
Instance Metadata and User Data	365
Stopping Instances	375
Stopping and Starting Instances	378
Terminating Instances	380
Enabling Termination Protection	381
Changing the Shutdown Behavior	384

Modifying Attributes of a Stopped Instance	386
Troubleshooting Instances	389
Instances with Failed Status Checks	389
What To Do If An Instance Immediately Terminates	412
Getting Console Output and Rebooting Instances	413
Network and Security	415
Security Groups	415
Controlling Access	421
Amazon VPC	428
Instance IP Addressing	430
Elastic IP Addresses	438
Elastic Network Interfaces	442
Storage	454
Amazon EBS	456
Increasing EBS Performance	461
Creating or Restoring a Volume	467
Attaching a Volume to an Instance	469
Describing Volumes	473
Making a Volume Available for Use	475
Monitoring the Status of Your Volumes	478
Detaching a Volume from an Instance	489
Deleting a Volume	491
Creating a Snapshot	492
Copying a Snapshot	494
Describing Snapshots	497
Modifying Snapshot Permissions	498
Deleting a Snapshot	501
Expanding a Volume	502
Updating a Linux Instance	505
Updating a Windows Instance	506
Using Public Data Sets	509
API and Command Overview	511
Instance Store	513
Amazon S3	521
Block Device Mapping	522
Resources and Tags	534
Resource Locations	534
Listing and Filtering Your Resources	535
Tagging Your Resources	540
Setting Up the CLI Tools	560
Verify the Signature	565
Making API Requests	570
Query Requests	571
SOAP Requests	575
Document History	576

What is Amazon EC2?

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable computing capacity in the cloud. Amazon EC2 eliminates your need to invest in hardware up front and enables you to develop and deploy applications faster. You can use the web service interfaces to launch as many or as few virtual servers as you need, configure security and networking, and manage storage.

With Amazon EC2, you pay only for the resources that you use. Amazon EC2 enables you to scale to handle your changes in requirements or spikes in popularity, reducing your need to forecast traffic.

Topics

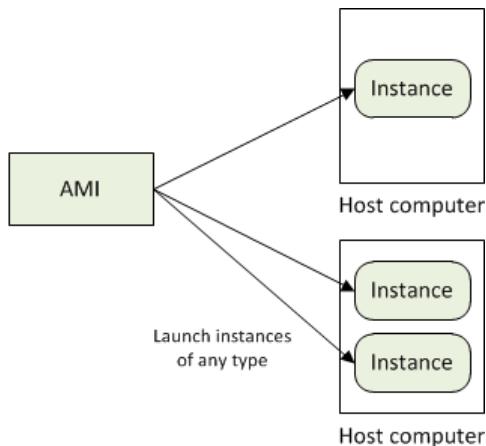
- [Amazon EC2 Concepts \(p. 1\)](#)
- [Working with Amazon EC2 \(p. 5\)](#)
- [How You're Charged for Amazon EC2 \(p. 6\)](#)
- [Amazon EC2 Resources \(p. 6\)](#)
- [What's Next? \(p. 7\)](#)

Amazon EC2 Concepts

As you get started with Amazon EC2, you should understand the key concepts of this computing environment and how it's similar to or different from your own computing environment.

Amazon Machine Images and Instances

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



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

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

You can launch different types of instances from a single AMI. An *instance type* essentially determines the hardware of the host computer used for your instance. Each instance type offers different compute and memory capabilities. Select an instance type based on the amount of memory and computing power that you need for the application or software that you plan to run on the instance. For more information, see [Available Instance Types \(p. 88\)](#).

When you launch an instance, the root device contains the image used to boot the instance. For more information, see [Amazon EC2 Root Device Volume \(p. 123\)](#).

All AMIs are categorized as either *backed by Amazon EBS*, which means that the root device for an instance launched from the AMI is an Amazon EBS volume, or *backed by instance store*, which means that the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3.

The description of an AMI indicates the type of root device (either `ebs` or `instance store`). This is important because there are significant differences in what you can do with each type of AMI. For more information about these differences, see [Storage for the Root Device \(p. 19\)](#).

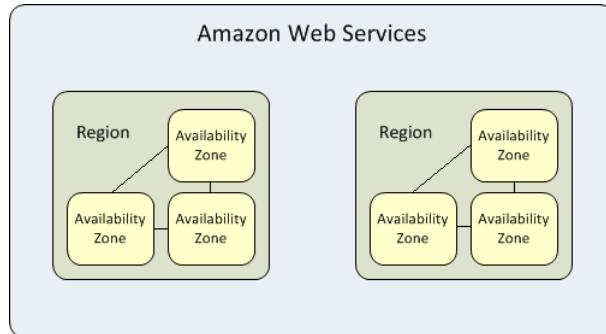
For information about AMIs and instances, see [Amazon Machine Images \(AMI\) \(p. 18\)](#) and [Amazon EC2 Instances \(p. 86\)](#).

Regions and Availability Zones

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

Each region contains multiple distinct locations called *Availability Zones*. Each Availability Zone is engineered to be isolated from failures in other Availability Zones, and to provide inexpensive, low-latency

network connectivity to other zones in the same region. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location.



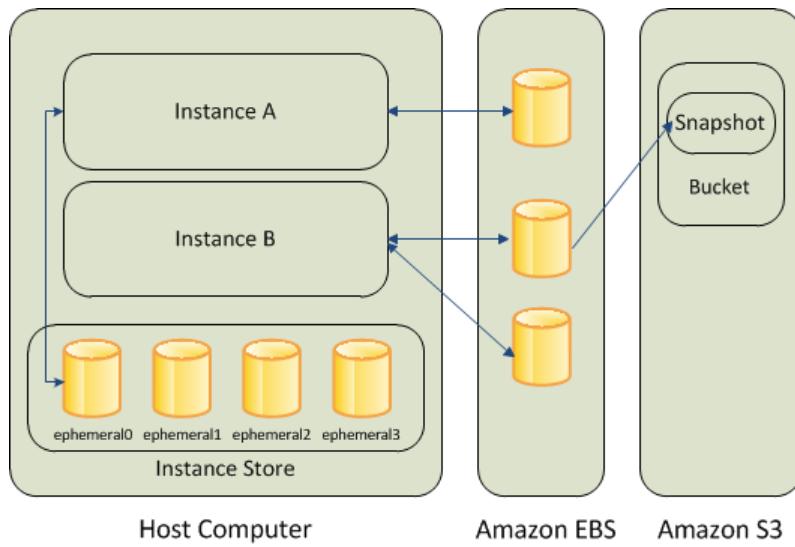
For more information about the available regions and Availability Zones, see [Regions and Availability Zones \(p. 116\)](#).

Storage

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

- Amazon Elastic Block Store (Amazon EBS)
- Amazon EC2 Instance Store (p. 513)
- Amazon Simple Storage Service (Amazon S3)

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



Amazon EBS

Amazon EBS volumes are the recommended storage option for the majority of use cases, and are especially suited for applications that require a database, a file system, or access to raw block-level storage. Amazon EBS volumes are essentially hard disks that you can attach to a running instance and provide your instance with persistent, block-level storage.

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

For more information about Amazon EBS volumes, see [Amazon Elastic Block Store \(Amazon EBS\) \(p. 456\)](#).

Instance Store

All instance types, with the exception of micro instances, offer *instance store*, which provides your instances with temporary, block-level storage. This is storage that is physically attached to the host computer. The data on an instance store volume doesn't persist when the associated instance is stopped or terminated. For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 513\)](#).

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

Amazon S3

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

Databases

The application you're running on Amazon EC2 may need a database. The following are common ways to implement a database for your application:

- Use Amazon Relational Database Service (Amazon RDS), which enables you to easily get a managed relational database in the cloud
- Launch an instance of a database AMI, and use that instance as the database

Amazon RDS offers the advantage of handling your database management tasks, such as patching the software, backing up, and storing the backups. For more information, see the [Amazon RDS product page](#).

Networking and Security

You can launch an instance into one of two platforms: EC2-Classic or EC2-VPC. An instance that's launched into EC2-Classic is assigned a public IP address. An instance that's launched into EC2-VPC is assigned a public IP address only if it's launched into a default VPC. For more information about EC2-Classic and EC2-VPC, see [Supported Platforms](#) in the *Amazon Elastic Compute Cloud User Guide*.

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

You can use *security groups* to control who can access your instances. These are analogous to an inbound network firewall that enables you to specify the protocols, ports, and source IP ranges that are allowed to reach your instances. You can create multiple security groups and assign different rules to each group. You can then assign each instance to one or more security groups, and we use the rules to determine which traffic is allowed to reach the instance. You can configure a security group so that only specific IP addresses or specific security groups have access to the instance. For more information about security groups, see [Amazon EC2 Security Groups \(p. 415\)](#).

Monitoring, Auto Scaling, and Load Balancing

AWS provides features that enable you to do the tasks described in the following table.

Task	Relevant Guide
Monitor basic statistics for your instances and Amazon EBS volumes.	Monitoring Your Instances with CloudWatch (p. 343)
Automatically scale your Amazon EC2 capacity up or down according to the conditions that you define.	Auto Scaling Developer Guide
Automatically distribute incoming application traffic across multiple EC2 instances.	Elastic Load Balancing Developer Guide

Working with Amazon EC2

AWS provides several interfaces for accessing Amazon EC2.

Topics

- [AWS Management Console \(p. 5\)](#)
- [Command Line Interface \(API Tools\) \(p. 5\)](#)
- [Programmatic Interfaces \(p. 5\)](#)

AWS Management Console

The AWS Management Console is a web-based user interface.

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

To get started, see [Getting Started with Amazon EC2 Linux Instances \(p. 8\)](#).

Command Line Interface (API Tools)

The command line interface for Amazon EC2 provides a set of commands that use a Java runtime environment. To get started, see [Setting Up the Amazon EC2 Command Line Interface Tools on Linux/UNIX \(p. 560\)](#). For more information about the commands, see [Amazon Elastic Compute Cloud Command Line Reference](#).

Programmatic Interfaces

The following table lists the resources that you can use to access Amazon EC2 programmatically.

Resource	Description
AWS SDKs	AWS SDKs include sample code, libraries, tools, documentation, and templates. To download the AWS SDKs, see AWS Software Development Kits (SDKs) .

Resource	Description
Libraries	Developers can provide their own libraries, which you can find at the following AWS developer centers: <ul style="list-style-type: none">• Java Developer Center• Mobile Developer Center• PHP Developer Center• Python Developer Center• Ruby Developer Center• Windows and .NET Developer Center
EC2 API	If you prefer, you can code directly to the Amazon EC2 API. For more information, see Making API Requests (p. 570) , and the Amazon Elastic Compute Cloud API Reference .

How You're Charged for Amazon EC2

With Amazon EC2, you pay for only what you use, and there's no minimum charge. For a complete list of charges and specific prices, see [Amazon EC2 pricing](#).

To see your bill, go to your [AWS Account Activity page](#). Your bill contains links to usage reports that provide details about your bill.

Amazon EC2 Resources

The following table lists related resources that you'll find useful as you work with Amazon EC2.

Resource	Description
AWS Resources	A central starting point to find documentation, code samples, release notes, and other information to help you create innovative applications using AWS. You can access this page from the Amazon EC2 console by clicking Help in the navigation bar, and selecting Other Resources .
Amazon EC2 forum	A community-based forum for discussing technical questions related to Amazon EC2. You can access the forum from the Amazon EC2 console by clicking Forums underneath Additional Information .
Amazon EC2 Release Notes	A high-level overview of the current release.
AWS Support Center	The home page for AWS Technical Support. You can access the Support Center from the Amazon EC2 console by clicking Help in the navigation bar, and selecting Support .

Resource	Description
AWS Support	The primary web page for information about AWS Support, a one-on-one, fast-response support channel to help you build and run applications on AWS.
AWS Trusted Advisor	An overview of Trusted Advisor, a service that helps you optimize the costs, security, and performance of your AWS environment. You can access the Trusted Advisor service from the Amazon EC2 console by clicking the AWS Trusted Advisor link underneath Resources .
Contact Us	A central contact point for inquiries concerning AWS billing, accounts, and events. This form is for account questions only. For technical questions, use the Discussion Forums.

You can also find additional information about Amazon EC2 in the [Amazon EC2 Articles & Tutorials](#) area of the AWS website.

What's Next?

Get a Hands-On Introduction to Amazon EC2

To get a hands-on introduction to Amazon EC2, complete the tutorial [Getting Started with Amazon EC2 Linux Instances \(p. 8\)](#).

Understand Differences Between the Cloud and Your Data Center

You need to understand the key differences between running your application on infrastructure in the cloud versus on servers in your own data center. For more information, go to the technical whitepaper: [Architecting for the Cloud: Best Practices](#).

Start Thinking about Instance Failure and Fault Tolerance

It's important for you to design your application to handle the failure of a cloud infrastructure component. For example, it's inevitable that instances will eventually fail, so you need to plan for it. An instance failure isn't a problem if your application is designed to handle it. For more information, see the technical whitepaper: [Building Fault-Tolerant Applications on AWS](#).

For a complete list of the AWS whitepapers, see [AWS Cloud Computing Whitepapers](#).

Within this guide, you should read the following sections to understand features of Amazon EC2 that help you build fault tolerant applications:

- [Elastic IP Addresses \(EIP\) \(p. 438\)](#)
- [Amazon Elastic Block Store \(p. 456\)](#)
- [Auto Scaling and Load Balancing Your Instances \(p. 340\)](#)
- [Monitoring Your Instances with CloudWatch \(p. 343\)](#)

Getting Started with Amazon EC2 Linux Instances

To get started using Amazon Elastic Compute Cloud (Amazon EC2) Linux instances, complete the steps shown in the following table. You'll primarily use the AWS Management Console, a point-and-click web-based interface. You can also watch this short video to get started: [How to Create a Red Hat Enterprise Linux EC2 Instance](#).

Tip

This tutorial has you launch and connect to an Amazon EC2 Linux instance. If you'd prefer to launch and connect to an Amazon EC2 Windows instance, see this tutorial: [Getting Started with Amazon EC2 Windows Instances](#).

For more information about Amazon EC2, see the [Amazon EC2 product page](#).

To get started with Amazon EC2

Step	Description
Step 1: Sign Up for Amazon EC2 (p. 9)	Create an AWS account, if you haven't done so already.
Step 2: Launch an Amazon EC2 Instance (p. 9)	Launch an Amazon Linux instance, which is a virtual server in the cloud.
Step 3: Connect to Your Instance (p. 13)	Connect to your instance.
Step 4: Explore Your Instance (p. 14)	Create an Amazon EC2 volume, attach it to your instance, and use Linux commands on your instance to make the volume available for use.
Step 5: Clean Up (p. 16)	Terminate your instance and delete the volume that you created.

Step 1: Sign Up for Amazon EC2

When you create an Amazon Web Services (AWS) account, AWS automatically signs up the account for all AWS services, including Amazon EC2.

With Amazon EC2, you pay only for what you use. New AWS customers can get started with Amazon EC2 for free. For more information, see [Amazon EC2 Pricing](#).

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

To create an AWS account

1. Go to <http://aws.amazon.com> and click **Sign Up Now**.
2. Follow the on-screen instructions.
Part of the sign-up process involves receiving a phone call and entering a PIN using the phone keypad.

Step 2: Launch an Amazon EC2 Instance

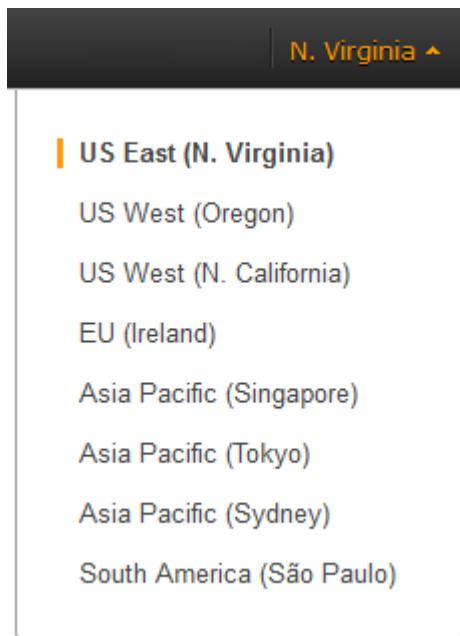
Now that you're signed up for AWS, you're ready to start "computing" in the cloud. The first thing you'll do is to launch a Linux instance using the AWS Management Console. An instance is a virtual server in the cloud. You can use Amazon EC2 to set up and configure the operating system and applications that run on your instance.

You can choose to launch one of the following instances:

- An instance within the Free Usage Tier. The Free Usage Tier enables you to launch and use an Amazon EC2 micro instance free for 12 months. For more information about the Free Usage Tier, see the [AWS Free Usage Tier product page](#) and [Getting Started with AWS Free Usage Tier](#).
- A regular instance (not within the Free Usage Tier). During this tutorial you'll incur the standard Amazon EC2 usage fees for the instance, until we show you how to terminate it in the last step. The total charges to complete this tutorial are minimal (typically less than a few dollars). For more information about Amazon EC2 usage fees, see [Amazon EC2 Pricing](#).

To launch an instance

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
Use the email address and password that you specified when signing up for AWS.
2. From the navigation bar, select the region for the instance. For this tutorial, you can use the default region. Otherwise, this choice is important because some EC2 resources can be shared between regions, while others can't. For example, if you'd like to connect your instance to an existing EBS volume, you must launch the instance in the same region as the volume.



3. From the Amazon EC2 console dashboard, click **Launch Instance**.

The **Create a New Instance** page includes these ways to launch an instance:

- The **Classic Wizard** offers you precise control and advanced settings for configuring your instance.
- The **Quick Launch Wizard** automatically configures many selections for you, so that you can get started quickly. This tutorial guides you through the Quick Launch Wizard.

4. On the **Create a New Instance** page, click **Quick Launch Wizard**.
5. (Optional) In **Name Your Instance**, enter a name for the instance that has meaning for you. (If you run multiple instances, naming them helps you identify them in the console.)
6. In **Choose a Key Pair**, you can choose from any existing key pairs that you've created, or you can create a new key pair.

A key pair enables you to connect to a Linux instance through SSH. Therefore, don't select the **None** option. If you launch your instance without a key pair, then you can't connect to it.

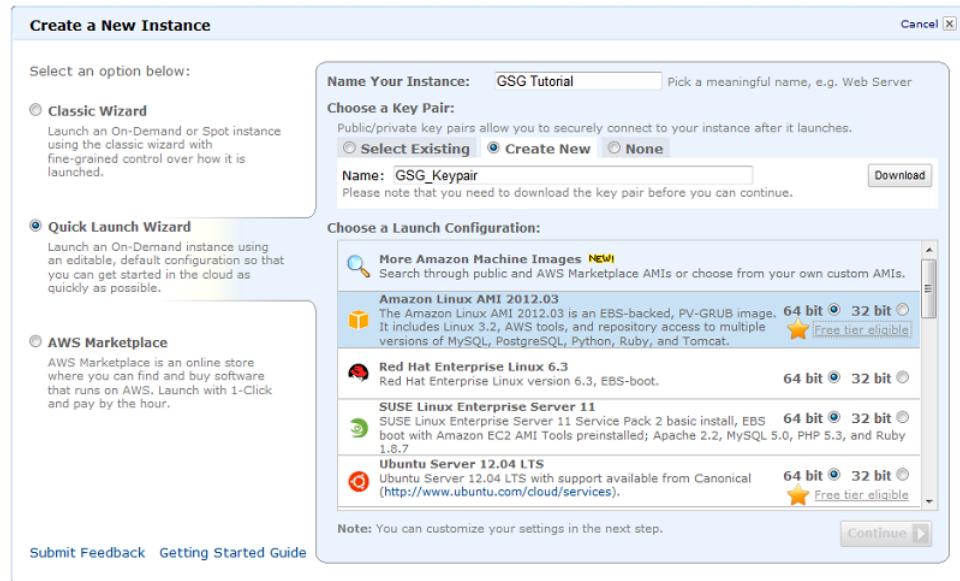
For this example, we'll create a key pair:

- a. Click **Create New**.
- b. Type a name for your key pair and then click **Download**.
- c. Save your private key in a safe place on your computer. Note the location because you'll need the key to connect to your instance.

7. In **Choose a Launch Configuration**, the Quick Launch Wizard displays a list of basic configurations called Amazon Machine Images (AMIs) that you can choose from to launch your instance. An AMI contains everything you need to create a new instance of a server, such as a web server or a database server. In this tutorial, we'll launch an Amazon Linux instance with a 64-bit operating system. If the configuration is marked with a star, this indicates that it's within the [Free Usage Tier](#).

Important

If you launch a regular instance, you're billed from the time that you launch the instance until the instance is stopped or terminated, even if it remains idle.



8. Click **Continue** to view and customize the settings for your instance.
9. In **Security Details, Security Group**, you see the security group that the wizard selects for you.

A security group defines firewall rules for your instances. These rules specify which incoming network traffic is delivered to your instance. All other traffic is ignored.

If you're new to Amazon EC2 and haven't set up any security groups yet, AWS defines a default security group for you. The name and description for the group is quicklaunch-x where x is a number associated with your quicklaunch group. The first security group you create using the Quick Launch Wizard is named quicklaunch-1. You can change the name and description using the **Edit details** button. The group already has basic firewall rules that enable you to connect to the type of instance you choose. For a Linux instance, you connect through SSH on port 22. The quicklaunch-x security group automatically allows SSH traffic on port 22.

If you have used Amazon EC2 before, the wizard looks for an existing security group for the type of instance you're creating.

Caution

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

Create a New Instance

Amazon Linux AMI 2012.03 (ami-aecd60c7)
 Platform: Amazon Linux
 Architecture: x86_64

Please review your settings and click **Launch** to finish or **Edit details** to make changes.

Instance Details Name: GSG Tutorial Detailed Monitoring: No Shutdown Behaviour: Stop Launch into: Default Subnet in any AZ	Type: t1.micro Availability Zone: No preference Termination Protection: No
Security Details Key Pair: GSG_Keypair Security Group: quicklaunch-1	
Advanced Details Kernel ID: Default Ramdisk ID: Default User Data: IAM Role:	

[◀ Go Back](#) [Edit details](#) **Launch ➔**

10. Review your settings, and click **Launch** to launch the instance.
11. A confirmation page lets you know that your instance is launching. Click **Close** to close the confirmation page and return to the Amazon EC2 console.
12. Click **Instances** in the navigation pane to view the status of your instance. It takes a short time for an instance to launch. The instance's status is pending while it's launching.

	Name	Instance	AMI ID	Root Device	Type	State
	GSG Tutorial	i-e1ab569a	ami-aecd60c7	ebs	t1.micro	pending

After the instance is launched, its status changes to running.

	Name	Instance	AMI ID	Root Device	Type	State
	GSG Tutorial	i-e1ab569a	ami-aecd60c7	ebs	t1.micro	running

13. (Optional) After your instance is launched, you can view the quicklaunch-x security group rules.
 - a. On the Amazon EC2 console, in **Network and Security**, click **Security Groups**.
 - b. Click the quicklaunch-1 security group to view the security rules created by the Quick Launch Wizard.

The screenshot shows the AWS Management Console interface for managing security groups. The title bar says "Security Group: quicklaunch-1". Below it, there are three tabs: "Details", "Inbound" (which is selected), and "Outbound".

Under "Create a new rule:", the dropdown menu is set to "Custom TCP rule".
The "Port range:" field contains "(e.g., 80 or 49152-65535)".
The "Source:" field contains "0.0.0.0/0" (e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default).
A green "Add Rule" button is located below the source field.

On the right, a table lists the existing rule:

TCP Port (Service)	Source	Action
22 (SSH)	0.0.0.0/0	Delete

A grey "Apply Rule Changes" button is at the bottom.

The security group contains a rule that authorizes SSH traffic from any IP address source to port 22. If you launch a Linux instance running Apache and MySQL, the Quick Launch Wizard creates a security group that authorizes traffic to port 80 for HTTP (for web traffic) and port 3306 (for MySQL).

Step 3: Connect to Your Instance

There are several ways to connect to a Linux instance. In this step, we show you how to connect to your instance using your browser. Alternatively, there are tools you can use to connect to your instance instead of connecting using your browser. For more information, see the following topics: [Connect to Linux/UNIX Instances from Linux/UNIX with SSH](#) and [Connecting to Linux/UNIX Instances from Windows Using PuTTY](#).

To connect to your instance through a web browser

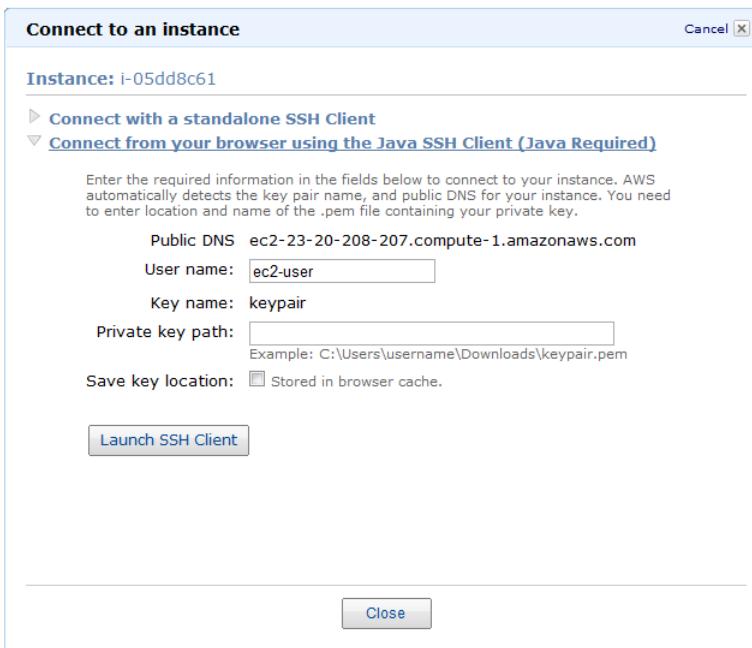
1. You must have Java installed and enabled in the browser. If you don't have Java already, you can contact your system administrator to get it installed, or follow the steps outlined in these pages:
 1. [Install Java](#)
 2. [Enable Java in your web browser](#)

2. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
3. Click **Instances** in the navigation pane.
4. Right-click your instance, and then click **Connect**.
5. Click **Connect from your browser using the Java SSH client (Java Required)**. AWS automatically detects the public DNS address of your instance and the key pair name you launched the instance with.
6. In **User name**, enter the user name to log in to your instance.

Note

For an Amazon Linux instance, the default user name is `ec2-user`. For Ubuntu, the default user name is `ubuntu`. Some AMIs allow you to log in as `root`. So in this example, you may need to change the user name from `ec2-user` to the appropriate user name.

7. The **Key name** field is automatically populated for you.
8. In **Private key path**, enter the fully qualified path to your `.pem` private key file.
9. Click **Save key location**, and then click **Stored in browser cache** to store the key location in your browser cache. This ensures that the key location will be detected in subsequent browser sessions, until you clear your browser's cache.
10. Click **Launch SSH Client**.



11. When prompted to add the host to your set of known hosts, click **No**.
12. If necessary, click **Yes** to trust the certificate.
13. Click **Run** to run the MindTerm client.
14. If you accept the license agreement, click **Accept**.
15. If this is your first time running MindTerm, a series of dialog boxes ask you to confirm setup for your home directory and other settings.
16. Confirm settings for MindTerm setup. A screen opens and you are connected to your instance.

Congratulations! You've successfully launched and connected to a Linux instance. If you'd like to explore the instance, continue to the next step. Otherwise, skip to [Step 5: Clean Up \(p. 16\)](#) to terminate the instance so that you don't continue to incur charges.

Step 4: Explore Your Instance

Now that you've connected to your instance, you can use it in the way that you'd use a Linux computer sitting in front of you.

Let's take a look at the default storage for the instance. In MindTerm, run the following command to view the mounted volumes.

```
$ df -h
```

For a micro instance, your output should look something like this.

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/xvda1	8.0G	1.1G	6.9G	14%	/
tmpfs	298M	0	298M	0%	/dev/shm

The /dev/xvda1 volume is the root device volume. It contains the image used to boot the instance.

Notice that there's room to install additional software on your instance. For example, you can use **yum** to download and install packages on your instance.

If you need additional storage for data, a simple solution is to add Amazon Elastic Block Store (Amazon EBS) volumes to your Amazon EC2 instance. An Amazon EBS volume serves as network-attached storage for your instance. Let's add a volume to the instance that you've launched. First we'll use the EC2 console to create an Amazon EBS volume and attach it to the instance, and then we'll connect to the instance and mount the volume.

To create and attach an Amazon EBS volume

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Select the region in which you created your instance in the navigation bar, and then click **Instances** in the navigation pane.
The console displays the list of current instances in that region. Select your Linux instance. In the **Description** tab in the bottom pane note the **Zone** for the instance.
3. In the navigation pane, in **Elastic Block Store**, click **Volumes**.
The console displays a list of current EBS volumes in that region. (You should see the EBS volume that serves as the root device volume for your instance.)
4. Click **Create Volume**.
5. In the **Create Volume** dialog box, configure the following settings:
 - Leave the volume size blank. (We'll let the snapshot we select determine the size of the volume.)
 - Select the same **Availability Zone** that you used when you created your instance. Otherwise, you can't attach the volume to your instance.
 - Select a snapshot that contains a public data set hosted by AWS. This enables us to quickly and easily create a volume that is formatted and contains data. The Free Usage Tier provides up to 30 GB of Amazon Elastic Block Storage; therefore, select a public data set that is smaller than 30 GB. We will use snap-2767d046 -- This is a 10,000 song..., which contains the [Million Song Sample Dataset](#). This data set is 5 GB in size. The snapshot list can contain a lot of snapshots. To quickly and easily find the Million Song Sample Dataset snapshot, select the **Snapshot** dropdown list and start typing the snapshot ID snap-2767d046 until the proper snapshot is found.
 - Select the **Standard** volume type. This creates a standard EBS volume.

Now click **Yes, Create**. This creates a new volume (its state is `available`, and its snapshot is `snap-2767d046`).

6. Right-click the newly created volume and select **Attach Volume**.
7. In the **Attach Volume** dialog box, click the following settings:
 - Select your Linux instance from the list.
 - Specify the device name /dev/sdf.

Now click **Yes, Attach**. You'll notice in the **Details** pane for your volume that the volume state is `in-use` and that it is attached to your instance with the device name /dev/sdf. However, if you return to MindTerm and run the `df -h` command again, you won't see the volume yet. That's because we need to mount the volume to make it available. To mount a volume based on a snapshot, run the following commands.

```
$ sudo mkdir /mnt/song-data
$ sudo mount /dev/sdf /mnt/song-data
```

Note

If you didn't specify a snapshot when you created the volume, the new volume is an empty volume. Before you can mount an empty volume that you create, you must format it using the **sudo mkfs** command. For more information, see [Making an Amazon EBS Volume Available for Use](#).

Now when you run the **df -h** command, you'll see output like the following.

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/xvda1	8.6G	1.2G	7.4G	14%	/
tmpfs	313M	0	313M	0%	/dev/shm
/dev/xvdः	5.0G	4.3G	442M	91%	/mnt/song-data

To view the contents of the new volume, run the following command.

```
$ dir /mnt/song-data
```

The output shows the following directory structure.

```
AdditionalFiles  data  LICENSE  lost+found  README
```

At this point, you should have a basic understanding of instances and how you work with them.

When you're finished with your instance, don't forget to clean up any resources you've used and terminate the instance, as shown in the next step.

Step 5: Clean Up

Now that you've completed this tutorial, you can customize the instance to meet your needs and keep using it.

Important

Remember, if you launched an instance in the Free Usage Tier, there are no charges. However, if you launched a regular instance, as soon as your instance starts to boot, you're billed for each hour or partial hour that you keep the instance running, even if the instance is idle. You'll stop incurring charges for a regular instance as soon as the instance status changes to `shutting down` or `terminated`. For more information, see [Amazon EC2 Pricing](#).

Terminating an instance effectively deletes it because you can't reconnect to the instance after you've terminated it. This differs from stopping the instance; when you stop an instance, it is shut down and you are not billed for hourly usage or data transfer (but you are billed for any Amazon EBS volume storage). Also, you can restart a stopped instance at any time with a start request. Each time you transition an instance from stopped to started, we charge a full instance hour, even if these transitions happen multiple times within a single hour. For more information about the differences between stopping and terminating an instance, see [Stopping Instances](#).

To terminate the instance

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. Locate your instance in the list of instances on the **Instances** page.
3. Right-click the instance, and then click **Terminate**.
4. Click **Yes, Terminate** when prompted for confirmation.
Amazon EC2 begins terminating the instance.

Amazon EBS volumes can persist even after your instance goes away. If you created and attached an EBS volume in the previous step, it was detached when you terminated the instance. However, you must delete the volume, or you'll be charged for volume storage if the storage amount exceeds the limit of the Free Usage Tier. After you delete a volume, its data is gone and the volume can't be attached to any instance.

To delete the volume

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Locate the volume that you created in the list of volumes on the **Volumes** page.
3. Right-click the volume, and then click **Delete**.
4. Click **Yes, Delete** when prompted for confirmation.
Amazon EC2 begins deleting the volume.

Finding Additional Tutorials

This tutorial is focused on learning to use Amazon EC2. For tutorials that show you how to use additional AWS products and services with Amazon EC2, see [Getting Started with AWS Documentation](#).

Amazon Machine Images (AMI)

Topics

- [AMI Basics \(p. 18\)](#)
- [Finding a Suitable AMI \(p. 21\)](#)
- [Shared AMIs \(p. 24\)](#)
- [Paid AMIs \(p. 26\)](#)
- [Creating Amazon EBS-Backed AMIs Using the Console \(p. 28\)](#)
- [Creating Your Own AMIs \(p. 30\)](#)
- [Copying AMIs \(p. 67\)](#)
- [Amazon Linux AMIs \(p. 71\)](#)

Amazon Machine Images (AMIs) are the basic building blocks of Amazon EC2. An AMI is a template that contains a software configuration (operating system, application server, and applications) that you can run on Amazon's proven computing environment. Before you can accomplish anything with Amazon EC2, you must understand the concepts in this section.

Related Topics

- [Amazon EC2 Instances \(p. 86\)](#)
- [Tagging Your Amazon EC2 Resources \(p. 540\)](#)

AMI Basics

An AMI contains all information necessary to boot an Amazon EC2 instance with your software. An AMI is like a template of a computer's root volume. For example, an AMI might contain the software to act as a web server (Linux, Apache, and your web site) or it might contain the software to act as a Hadoop node (Linux, Hadoop, and a custom application). You launch one or more instances from an AMI. An instance might be one web server within a web server cluster or one Hadoop node.

Topics

- [Available AMIs \(p. 19\)](#)
- [Storage for the Root Device \(p. 19\)](#)

Available AMIs

Public AMIs can be launched by all AWS accounts. They are advertised and shared in the AWS AMI catalog. Amazon and the Amazon EC2 community provide a large selection of public AMIs. For more information, go to Amazon Machine Images (AMIs) in the AWS forum (<https://forums.aws.amazon.com/>).

Paid AMIs are AMIs that you purchase from the AWS Marketplace, developers, or through a service contract from an organization such as Red Hat. If you are interested in the AWS Marketplace, visit [AWS Marketplace](#). If you're interested in selling an AMI to other developers, go to <http://aws.amazon.com/devpay>.

The AWS Management Console (at <http://aws.amazon.com/console>) enables you to search for an AMI that meets specific criteria, and then launch instances from that AMI. For example, you can view the AMIs that Amazon has provided, the AMIs that the EC2 community has provided, or AMIs that use a certain platform (such as Windows, Red Hat, CentOS, or Ubuntu).

You might find that public AMIs suit your needs. However, you might want to customize a public AMI and save that customized AMI for your own use later on. For more information about creating your own AMIs, see [Creating Your Own AMIs \(p. 30\)](#).

After you create a new AMI, you can keep it private so that only you can use it, or you can share it with only specific AWS accounts that you specify. Another option is to make your customized AMI public so that the EC2 community can use it. Building safe, secure, usable AMIs for public consumption is a fairly straightforward process, if you follow a few simple guidelines. For information on how to use shared AMIs and how to share AMIs, see [Shared AMIs \(p. 24\)](#) and [Sharing AMIs Safely \(p. 58\)](#).

To help categorize and manage your AMIs, you can assign *tags* of your choice to them. For more information, see [Tagging Your Amazon EC2 Resources \(p. 540\)](#).

Storage for the Root Device

All AMIs are categorized as either *backed by Amazon EBS* or *backed by instance store*. The former means that the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot. The latter means that the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3. For more information, see [Amazon EC2 Root Device Volume \(p. 123\)](#).

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

Characteristic	Amazon EBS-Backed	Amazon instance store-backed
Boot Time	Usually less than 1 minute	Usually less than 5 minutes
Size Limit	1 TiB	10 GiB
Root Device Volume	Amazon EBS volume	Instance store volume
Data Persistence	Data on Amazon EBS volumes persists after instance termination; you can also attach instance store volumes that don't persist after instance termination	Data on instance store volumes persists only during the life of the instance; you can also attach Amazon EBS volumes that persist after instance termination
Upgrading	The instance type, kernel, RAM disk, and user data can be changed while the instance is stopped.	Instance attributes are fixed for the life of an instance

Characteristic	Amazon EBS-Backed	Amazon instance store-backed
Charges	Instance usage, Amazon EBS volume usage, and Amazon EBS snapshot charges for AMI storage	Instance usage and Amazon S3 charges for AMI storage
AMI Creation/Bundling	Uses a single command/call	Requires installation and use of AMI tools
Stopped State	Can be placed in stopped state where instance is not running, but the instance is persisted in Amazon EBS	Cannot be in stopped state; instances are running or terminated

Size Limit

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

Note

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

Stopped State

You can *stop* an Amazon EBS-backed instance, but not an Amazon EC2 instance store-backed instance. Stopping causes the instance to stop running (its status goes from *running* to *stopping* to *stopped*). A stopped instance persists in Amazon EBS, which allows it to be restarted. Stopping is different from terminating; you can't restart a terminated instance. Because Amazon EC2 instance store-backed AMIs can't be stopped, they're either running or terminated. For more information about what happens and what you can do while an instance is stopped, see [Stopping Instances \(p. 375\)](#).

Default Data Storage and Persistence

Instances that use instance store for the root device automatically have instance stores available on them (for the root partition and other data you want to add). Any data on those stores is deleted if the instance fails or terminates (except for data on the root device). However, after you launch an instance, you can provide it with persistent non-root data by attaching one or more Amazon EBS volumes.

Instances that use Amazon EBS for the root device automatically have an Amazon EBS volume attached. The volume appears in your list of volumes like any other. The instances don't expose ephemeral storage by default. However, you can add ephemeral storage by using a block device mapping. You can also add additional Amazon EBS volumes for non-root data; for more information, see [Block Device Mapping \(p. 522\)](#)). For information about what happens to the ephemeral storage and volumes when you stop an instance, see [Stopping Instances \(p. 375\)](#).

Boot Times

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

AMI Creation

To create Linux/UNIX AMIs backed by instance store, you must create an image of your instance on the instance itself, but there aren't any API actions to help you. To create a Windows AMI backed by instance store, there's an API action that creates an image and another API action that registers the AMI.

AMI creation is much easier for AMIs backed by Amazon EBS. The `CreateImage` API action creates the AMI on both Linux/UNIX and Windows. This API action creates your Amazon EBS-backed AMI and registers it. There's also a button in the AWS Management Console that lets you create an image from a running instance. For more information, see [Creating Amazon EBS-Backed Linux AMIs \(p. 31\)](#).

How You're Charged

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

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

When an Amazon EBS-backed instance is stopped, you're not charged for instance usage; however, you're still charged for volume storage. We charge a full instance hour for every transition from a stopped state to a running state, even if you transition the instance multiple times within a single hour. For example, let's say the hourly instance charge for your instance is \$0.10. If you were to run that instance for one hour without stopping it, you would be charged \$0.10. If you stopped and restarted that instance twice during that hour, you would be charged \$0.30 for that hour of usage (the initial \$0.10, plus 2 x \$0.10 for each restart).

Finding a Suitable AMI

This topic describes how to find an AMI that meets your needs.

AWS Management Console

To find a suitable AMI

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **AMIs**.
The console displays any AMIs that you own.
3. To change which AMIs are displayed, select options from the **Filter** drop-down lists. You can see all the available options by expanding the drop-down lists at the same time. This enables you to list the types of AMIs that interest you. For example, select **Public Images** then **Amazon Images** from the drop-down lists to display only Amazon's public images.
4. As you are selecting an AMI, it's important to note whether the AMI is backed by instance store or by Amazon EBS. To verify the type of root device volume that is used by an AMI, check the value in the **Root Device Type** column (`ebs` or `instance-store`).

For more information about the differences between these AMI types, see [Storage for the Root Device \(p. 19\)](#).

Note

You can also inspect the details of an AMI by clicking the **Go to Details Page** button (the magnifying glass) for that AMI. This takes you to an information page with the AMI's details, permissions and tags.

5. After locating an AMI that meets your needs, write down its AMI ID, which has the form ami-xxxxxxx. You can use this ID to launch instances of the AMI or register your own AMI, using this one as a baseline.

Command Line Tools

To find a suitable AMI

1. Use the [ec2-describe-images](#) command to list your AMIs and Amazon's public AMIs.

```
PROMPT> ec2-describe-images -o self -o amazon
```

The following example shows only part of the resulting output from the command (information for 10 AMIs).

```
IMAGE ami-d8699bb1 amazon/ami-vpc-nat-1.0.0-beta.i386-ebs amazon available
public i386 machine aki-407d9529 ebs paravirtual xen
BLOCKDEVICEMAPPING /dev/sda1 snap-33d88c5f 8
IMAGE ami-c6699baf amazon/ami-vpc-nat-1.0.0-beta.x86_64-ebs amazon available
public x86_64 machine aki-427d952b ebs paravirtual xen
BLOCKDEVICEMAPPING /dev/sda1 snap-57d88c3b 8
IMAGE ami-30f30659 amazon/amzn-ami-0.9.7-beta.i386-ebs amazon available
public i386 machine aki-407d9529 ebs paravirtual xen
BLOCKDEVICEMAPPING /dev/sda1 snap-d895cdb3 10
IMAGE ami-0af30663 amazon/amzn-ami-0.9.7-beta.x86_64-ebs amazon available
public x86_64 machine aki-427d952b ebs paravirtual xen
BLOCKDEVICEMAPPING /dev/sda1 snap-f295cd99 10
IMAGE ami-3ac33653 amazon/amzn-ami-0.9.8-beta.i386-ebs amazon available
public i386 machine aki-407d9529 ebs paravirtual xen
BLOCKDEVICEMAPPING /dev/sda1 snap-14ba967f 10
IMAGE ami-38c33651 amazon/amzn-ami-0.9.8-beta.x86_64-ebs amazon available
public x86_64 machine aki-427d952b ebs paravirtual xen
BLOCKDEVICEMAPPING /dev/sda1 snap-10b9957b 10
IMAGE ami-08728661 amazon/amzn-ami-0.9.9-beta.i386-ebs amazon available
public i386 machine aki-407d9529 ebs paravirtual xen
BLOCKDEVICEMAPPING /dev/sda1 snap-674a930d 10
IMAGE ami-2272864b amazon/amzn-ami-0.9.9-beta.x86_64-ebs amazon available
public x86_64 machine aki-427d952b ebs paravirtual xen
BLOCKDEVICEMAPPING /dev/sda1 snap-8926ffe3 10
IMAGE ami-76f0061f amazon/amzn-ami-2010.11.1-beta.i386-ebs amazon available
public i386 machine aki-407d9529 ebs paravirtual xen
BLOCKDEVICEMAPPING /dev/sda1 snap-cba692a1 8
IMAGE ami-74f0061d amazon/amzn-ami-2010.11.1-beta.x86_64-ebs amazon available
public x86_64 machine aki-427d952b ebs paravirtual xen
BLOCKDEVICEMAPPING /dev/sda1 snap-ffa69295 8
IMAGE ami-8c1fce5 amazon/amzn-ami-2011.02.1.i386-ebs amazon available
public i386 machine aki-407d9529 ebs paravirtual xen
BLOCKDEVICEMAPPING /dev/sda1 snap-22fc264e 8
IMAGE ami-8e1fce7 amazon/amzn-ami-2011.02.1.x86_64-ebs amazon available
public x86_64 machine aki-427d952b ebs paravirtual xen
BLOCKDEVICEMAPPING /dev/sda1 snap-a6fc26ca 8
```

2. To reduce the number of displayed AMIs, use a filter to list only the types of AMIs that interest you. For example, use `--filter "platform=windows"` to display only Windows-based AMIs.
3. After locating an AMI that meets your needs, write down its AMI ID, which has the form `ami-xxxxxxxx`. You can use this ID to launch instances of the AMI or register your own AMI, using this one as a baseline.

API

To find a suitable AMI

1. Use the [DescribeImages](#) action to list all Amazon AMIs. Construct the following request.

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

The following is an example response.

```
<DescribeImagesResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">  
  <imagesSet>  
    <item>  
      <imageId>ami-8c1fce5</imageId>  
      <imageLocation>amazon/amzn-ami-2011.02.1.i386-ebs</imageLocation>  
  
      <imageState>available</imageState>  
      <imageOwnerId>137112412989</imageOwnerId>  
      <isPublic>true</isPublic>  
      <architecture>i386</architecture>  
      <imageType>machine</imageType>  
      <kernelId>aki-407d9529</kernelId>  
      <imageOwnerAlias>amazon</imageOwnerAlias>  
      <name>amzn-ami-2011.02.1.i386-ebs</name>  
      <description>Amazon Linux AMI i386 EBS</description>  
      <rootDeviceType>ebs</rootDeviceType>  
      <rootDeviceName>/dev/sdal</rootDeviceName>  
      <blockDeviceMapping>  
        <item>  
          <deviceName>/dev/sdal</deviceName>  
          <ebs>  
            <snapshotId>snap-22fc264e</snapshotId>  
            <volumeSize>8</volumeSize>  
            <deleteOnTermination>true</deleteOnTermination>  
          </ebs>  
        </item>  
      </blockDeviceMapping>  
      <virtualizationType>paravirtual</virtualizationType>  
      <hypervisor>xen</hypervisor>  
    </item>  
  </imagesSet>
```

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

```
&Filter.1.Name=platform  
&Filter.1.Value.1=windows
```

3. After locating an AMI that meets your needs, write down its AMI ID, which has the form ami-xxxxxxx. You can use this ID to launch instances of the AMI or register your own AMI, using this one as a baseline.

Shared AMIs

This topic 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.

Topics

- [Find Shared AMIs \(p. 24\)](#)
- [Safe Use of Shared AMIs \(p. 25\)](#)

Find Shared AMIs

You can find a shared AMI using the console, the command line tools, or the API.

AWS Management Console

To find a shared AMI

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **AMIs**.
The console displays any AMIs you own.
3. To find shared AMIs, select **Public Images** from the **Filter** drop-down list.
4. To list only the type of AMIs that interest you, use the options from the **Filter** drop-down lists to filter on the type of image and the platform. For example, select **Amazon Images** to display only Amazon's public images.

Command Line Tools

To find shared AMIs, use the `ec2-describe-images` command with a flag to filter the results.

The following command displays a list of all public AMIs. The `-x all` flag shows AMIs that can be used by all AWS accounts to launch an instance (in other words, AMIs with public launch permissions). This includes the AMIs that you own with public launch permissions.

```
PROMPT> ec2-describe-images -x all
```

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

```
PROMPT> ec2-describe-images -x self
```

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

```
PROMPT> ec2-describe-images -o amazon
```

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

```
PROMPT> ec2-describe-images -o <target_uid>
```

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

To reduce the number of displayed AMIs, use a filter to list only the types of AMIs that interest you. For example, use `--filter "platform=windows"` to display only Windows-based AMIs.

API

To find a shared AMI

1. Use the [DescribeImages](#) action to list all Amazon AMIs. Construct the following request.

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

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

```
&Filter.1.Name=platform  
&Filter.1.Value.1=windows
```

Safe Use of Shared AMIs

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

Ideally, you should get the AMI ID from a trusted source (a website, another EC2 user, etc.). If you do not know the source of an AMI, we recommend 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 <https://forums.aws.amazon.com/> to ask or comment.

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

Note

Users cannot alias an AMI's owner.

If you plan to use a shared AMI, review the following table to confirm the instance 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 it is not randomized on startup. For more information on randomizing the root password on startup, see Disable Password-Based Logins for Root (p. 59) .
4	Check if SSH allows root password logins. For more information on disabling root based password logins, see Disable Password-Based Logins for Root (p. 59) and Restrict Access to the Root Account (p. 60) .
5	Check whether there are any other user accounts that might allow back-door entry to your instance. Accounts with super user privileges are particularly dangerous.
6	Verify that all cron jobs are legitimate.

For more information, see [Sharing AMIs Safely \(p. 58\)](#).

Paid AMIs

Topics

- [Find Paid AMIs \(p. 26\)](#)
- [Purchase a Paid AMI \(p. 27\)](#)
- [Launch Paid AMIs \(p. 27\)](#)
- [Using Paid Support \(p. 28\)](#)
- [Bills for Paid and Supported AMIs \(p. 28\)](#)

Amazon EC2 integrates with Amazon DevPay, enabling developers to charge other Amazon EC2 users for the use of their AMIs or to provide support for instances. For more information about Amazon DevPay, see the [Amazon DevPay Developer Guide](#). For more information about charging for the use of your AMIs, or providing support, see [Creating Paid AMIs \(p. 48\)](#).

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.

Find Paid AMIs

There are several ways you can find AMIs that 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 tell if an AMI is a paid AMI by describing the image using the console, the command line tools, or the API. If the AMI is a paid AMI, it has a product code; otherwise, it does not.

Note

You must sign up for a paid AMI before you can launch it. If you find a paid AMI that you are interested in, go to the Amazon EC2 resource center and forums, which might have more information about the paid AMI and where you can sign up to use it.

AWS Management Console

To find a paid AMI

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **AMIs**.
The console displays your AMIs.
3. To display the paid AMIs, select **Marketplace Images** from the middle **Filter** drop-down list.
To display the paid AMIs, select **Marketplace Images** from the middle **Filter** drop-down list.

Command Line Tools

To check whether an AMI is a paid AMI

1. Use the **ec2-describe-images** command as follows.

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

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

The command returns numerous fields that describe the AMI. If the output for the AMI contains a product code, the AMI is a paid AMI.

2. This example shows **ec2-describe-images** output for a paid AMI. The product code is ACD42B6F.

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

Purchase a Paid AMI

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

Typically a seller of a paid AMI presents you with information about the AMI, its price, and a link where you can buy it. When you click the link, you're first asked to log in with an Amazon.com login, and then you are taken to a page where you see the price for the paid AMI and can confirm that 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, see [Reserved Instances \(p. 212\)](#).

Launch Paid AMIs

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. For more information, see [Launching EC2 Instances \(p. 274\)](#).

The instance is charged according to the rates set by the owner of the AMI (which is more than the base Amazon EC2 rate).

Note

The owner of a paid AMI can confirm if a particular instance was launched using their paid AMI.

Using Paid Support

The paid AMI feature also enables 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 enables the developer to confirm that your instance is eligible for support. It also ensures that when you run instances of the product, you are charged according to the developer's terms for the product.

Important

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

To associate a product code with your AMI, use the [ec2-modify-image-attribute](#) command:

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

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

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
```

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, see [Reserved Instances \(p. 212\)](#).

Bills for Paid and Supported AMIs

At the end of each month, you receive an email with the amount your credit card has been charged for using the paid or supported AMIs during the month. This bill is separate from your regular Amazon EC2 bill.

At any time, you can view the usage information for your paid and supported AMIs; for more information, go to [Paid AMIs \(p. 26\)](#) and follow the instructions for how to view your paid and supported AMIs.

Creating Amazon EBS-Backed AMIs Using the Console

Topics

- [Create an AMI from an Instance \(p. 29\)](#)
- [Delete an AMI and a Snapshot \(p. 30\)](#)

This section walks you through the process of creating an Amazon EBS-backed AMI from a running Amazon EBS-backed instance. For more information about Amazon EBS-backed AMIs and instance store-backed AMIs, see [Storage for the Root Device \(p. 19\)](#). For instructions that use the command line tools or API, see [Creating Amazon EBS-Backed Linux AMIs \(p. 31\)](#).

Create an AMI from an Instance

To create an AMI from a running Amazon EBS-backed instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. If you don't have a running instance that uses an Amazon EBS volume for the root device, you must launch one. For instructions, see [Launching an Instance from an AMI \(p. 284\)](#).
3. [Optional] Connect to the instance and customize it however you want. For example, you can install software and applications, copy data, or attach additional EBS volumes. For more information about connecting to an instance, see [Connecting to Amazon EC2 Instances \(p. 293\)](#).
4. In the navigation pane, click **Instances** to view a list of your instances. Right-click your running instance and select **Create Image (EBS AMI)**.

Tip

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

The **Create Image** dialog box appears.

The screenshot shows the 'Create Image' dialog box. At the top, the instance ID is listed as i-ad019dd6. Below it, there are fields for 'Image Name*' and 'Image Description*', both currently empty. A checkbox for 'No Reboot:' is checked. Under volume selection, 'Root Volume' is selected, while 'EBS Volumes' and 'Instance Store Volumes' are unselected. A note says 'Optional edit the the root volume of your instance.' Below this, the 'Volume Size:' is set to 30 GiB, 'Volume Type:' is 'Standard', and 'IOPS:' is set to 100. A 'Delete on Termination:' checkbox is checked. On the right, a 'Save' button is checked. A table below lists the volume details: Type (Root), Device (/dev/sda1), Snapshot ID, Size (30GB), Volume Type (standard), and Delete on Termination (true). A note at the bottom states 'Total size of EBS volumes: 30 GB.' and 'When you create an EBS image an EBS snapshot will also be created for each of the above volumes.' At the bottom right are 'Cancel' and 'Yes, Create' buttons.

5. Fill in the requested information as follows, and click **Yes, Create**.
 - a. A unique name for the image.
 - b. [Optional] A description of the image (up to 255 characters).
 - c. By default, Amazon EC2 shuts down the instance, takes snapshots of any attached volumes, creates and registers the AMI, and then reboots the instance. Select **No Reboot** if you don't want your instance to be shut down.
- d. [Optional] You can modify the root volume, EBS volumes, and instance store volumes as follows:

Warning

If you select the **No Reboot** option, the file system integrity of the created image can't be guaranteed.

- To change the size of the root volume, click **Root Volume**, fill in the **Volume Size** field, and click **Save**.
- To suppress an EBS volume specified by the block device mapping of the AMI used to launch the instance, click **EBS Volumes**, go to the volume, and click **Delete**.
- To add an EBS volume, click **EBS Volumes**, fill in the fields (**Device**, **Snapshot**, and so on), and click **Add**. When you then launch an instance from your new AMI, these additional volumes are automatically attached to the instance. Empty volumes must be formatted and mounted. Volumes based on a snapshot must be mounted.
- To suppress an instance store volume specified by the block device mapping of the AMI used to launch the instance, click **Instance Store Volumes**, go to the volume, and click **Delete**.
- To add an instance store volume, click **Instance Store Volumes**, select the **Instance Store** and **Device**, and click **Add**. When you launch an instance from your new AMI, these additional volumes are automatically initialized and mounted. These volumes don't contain data from the instance store volumes of the running instance from which you based your AMI.

For more information, see [Amazon EC2 Root Device Volume \(p. 123\)](#), [Amazon EC2 Instance Store \(p. 513\)](#), and [Block Device Mapping \(p. 522\)](#)

6. Click **AMIs** in the navigation pane to view the AMI's status. While the new AMI is being created, its status is **pending**.
It takes a few minutes for the whole process to finish.
7. After your new AMI's status is **available**, go to the **Snapshots** page and view the new snapshot that was created for the new AMI. Any instance you launch from the new AMI uses this snapshot for its root device volume.
8. Go back to the **AMIs** page, select the image, and click **Launch**.
The launch wizard opens.
9. Walk through the wizard to launch an instance of your new AMI.
10. After your instance's status is **running**, connect to the instance and verify that any changes you made to the original AMI have persisted.

You now have a new AMI and snapshot that you just created. Both continue to incur charges to your account until you stop or delete them.

Delete an AMI and a Snapshot

To delete an AMI and a snapshot

1. Go to the **AMIs** page. Select the AMI, click **Actions**, then select **Deregister**. When asked for confirmation, click **Continue**.
The image is deregistered, which means it is deleted and can no longer be launched.
2. Go to the **Snapshots** page. Right-click the snapshot and select **Delete Snapshot**. When asked for confirmation, click **Yes, Delete**.
The snapshot is deleted.

Creating Your Own AMIs

There are many public AMIs available to you. To see the available AMIs, go to [Amazon Machine Images \(AMIs\)](#). If the available public AMIs don't provide everything that you're looking for, you can create an AMI that meets your needs.

Creating your own AMI helps you make the most of Amazon EC2. Your AMI becomes the basic unit of deployment; it enables you to rapidly boot new custom instances as you need them. This section gives an overview of your AMI creation options, identifies the tools you need, and walks you through the process.

Before you begin this section, you should be familiar with AMI and instance concepts. For more information, see the following sections:

- [Amazon Machine Images \(AMI\) \(p. 18\)](#)
- [Amazon EC2 Instances \(p. 86\)](#)
- [Amazon Elastic Block Store \(p. 456\)](#)

Overview of the AMI Creation Process

There are a few different ways to create an AMI. The process you must follow to create an AMI depends on whether you are creating an Amazon EBS-backed AMI or an Amazon EC2 instance store-backed AMI. There are significant differences between Amazon EBS-backed and Amazon EC2 instance store-backed AMIs, such as data persistence. For information on the differences between these choices, see [Storage for the Root Device \(p. 19\)](#).

First, decide which operating system and root device volume you want, and then you'll know which of these processes to use for creating the AMI:

- **Amazon EBS-backed AMI**
The same general process applies to Linux/UNIX and Windows.
 - [Linux/UNIX—Creating Amazon EBS-Backed Linux AMIs \(p. 31\)](#)
 - [Windows— Creating an Amazon EBS-Backed Windows AMI \(Amazon Elastic Compute Cloud Microsoft Windows Guide\)](#)
- **Amazon EC2 instance store-backed AMI**
The process depends on the operating system.
 - [Linux/UNIX—Creating Instance Store-Backed Linux/UNIX AMIs \(p. 34\)](#)
 - [Windows— Creating an Instance Store-Backed Windows AMI \(Amazon Elastic Compute Cloud Microsoft Windows Guide\)](#)

Creating Amazon EBS-Backed Linux AMIs

To create an Amazon EBS-backed Linux AMI, start with an Amazon EBS-backed AMI (for example, one of the public AMIs that Amazon provides), and modify it to suit your particular needs (note that as Amazon EBS-backed instances are stored as Amazon EBS data, standard storage rates apply). If you start with an Amazon instance store-backed instance, you cannot create an Amazon EBS-backed AMI using these instructions. For more information about Amazon EBS-backed AMIs and instance store-backed AMIs, see [Storage for the Root Device \(p. 19\)](#).

Note

This topic describes the process for creating an Amazon EBS-backed Linux AMI. For information about Amazon EBS-backed Windows AMIs, see [Creating an Amazon EBS-Backed Windows AMI](#). For instructions using an instance store-backed AMI, see [Creating Instance Store-Backed Linux/UNIX AMIs \(p. 34\)](#).

Topics

- [Creating an Amazon EBS-Backed Linux AMI \(p. 32\)](#)
- [Special Cases \(p. 32\)](#)
- [How to Create Amazon EBS-Backed AMIs \(p. 33\)](#)
- [Converting Amazon EC2 instance store-backed AMIs to EBS-Backed AMIs \(p. 34\)](#)

Creating an Amazon EBS-Backed Linux AMI

To create an Amazon EBS-backed Linux AMI

1. Launch an instance of an Amazon EBS-backed AMI that is similar to the AMI that you want to create. For example, you might take a public AMI that uses the operating system you want to use for your AMI.

The instance must be launched from an Amazon EBS-backed AMI; you cannot use an Amazon EC2 instance store-backed AMI.

2. When the instance is running, customize it as desired. For example, you could attach additional Amazon EBS volumes, load applications, or copy data.

Important

If you customize your instance with instance-store volumes or additional EBS volumes besides the root device, the new AMI contains block device mapping information for those volumes and new instances automatically launch with the additional volumes. However, instance-store volumes on the new instance won't contain any customized data. For more information, see [Create an AMI from an Instance](#). If you want your data to persist, you must use an EBS volume instead of an instance-store (ephemeral) volume. For more information, see [Block Device Mapping \(p. 522\)](#).

3. When the instance is set up the way you want it, it is best to stop the instance before you create the AMI to ensure data integrity. Follow these steps to stop the instance:
 - a. Right-click your running instance and select **Stop Instance**.
 - b. When prompted to confirm this, click **Yes, Stop Instance**.

4. Create an AMI from that instance.

It takes several minutes for the entire process to complete. If you customized the instance with instance store volumes or additional EBS volumes besides the root device, the new AMI contains block device mapping information for those volumes. When you launch an instance from your new AMI, the instance automatically launches with the additional volumes. The instance store volumes are new and don't contain any data from the instance store volumes of the original instance used to create the AMI.

Amazon EC2 powers down the instance before creating the AMI to ensure that everything on the instance is stopped and in a consistent state during the creation process. If you're confident that your instance is in a consistent state appropriate for AMI creation, you can add the `--no-reboot` flag to `ec2-create-image` or `CreateImage` that tells Amazon EC2 not to power down and reboot the instance. With this flag, the instance remains running throughout the AMI creation process. Some file systems, such as xfs, can freeze and unfreeze activity, making it safe to create the image without rebooting the instance.

Special Cases

In some cases, the general tasks in creating Amazon EBS-backed AMIs don't apply:

- You don't have the original AMI from which to launch instances.

In this case, you can create an Amazon EBS-backed AMI by registering a snapshot of a root device. You must own the snapshot and it must be a Linux/UNIX system (this process is not available for Windows instances). For more information about creating an AMI this way, see [Creating and Launching an AMI from a Snapshot \(p. 290\)](#).

- You have an Amazon EC2 instance store-backed Linux/UNIX AMI.

In this case, you can convert the AMI to be backed by Amazon EBS. You cannot convert a Windows AMI backed by instance store. For more information about converting a Linux/UNIX AMI, see [Converting Amazon EC2 instance store-backed AMIs to EBS-Backed AMIs \(p. 34\)](#).

How to Create Amazon EBS-Backed AMIs

You can create an Amazon EBS-backed AMI by using the AWS Management Console, the command line tools, or the API. The following section describes the steps using each tool or interface.

AWS Management Console

For instructions that use the AWS Management Console, see [Creating Amazon EBS-Backed AMIs Using the Console \(p. 28\)](#).

Command Line Interface

To create an Amazon EBS-backed AMI

1. Use the [ec2-create-image](#) command to create an image.

```
PROMPT> ec2-create-image -n your_image_name instance_id
```

For example:

```
PROMPT> ec2-create-image -n "My AMI" i-eb977f82
```

Amazon EC2 creates an image and returns an AMI ID.

```
IMAGE ami-8675309
```

2. If you want to check whether the AMI is ready, use the [ec2-describe-images](#) command as follows:

```
$ ec2-describe-images -o self
```

Amazon EC2 returns information about the AMI.

If the AMI you start with doesn't already have the storage devices you want attached, you can add them by creating EBS volumes or using block device mapping. To create EBS volumes, use [ec2-create-volume](#) and [ec2-attach-volume](#).

You also can call [ec2-run-instances](#) with block device mapping information for the devices you want to add. For more information about block device mapping, see [Block Device Mapping \(p. 522\)](#).

API

To create an Amazon EBS-backed AMI, construct the following query request to create an image:

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

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

```
<CreateImageResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
  <imageId>ami-8675309</imageId>
</CreateImageResponse>
```

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

If the AMI you start with doesn't already have the storage devices you want attached, you can add them by creating EBS volumes or using block device mapping. To create EBS volumes, use [CreateVolume](#) and [AttachVolume](#).

You also can call [RunInstances](#) with block device mapping information for the devices you want to add. For more information about block device mapping, see [Block Device Mapping \(p. 522\)](#).

Converting Amazon EC2 instance store-backed AMIs to EBS-Backed AMIs

There's no simple API or button in the AWS Management Console that converts an existing Amazon EC2 instance store-backed AMI to an Amazon EBS-backed AMI. However, you can convert Amazon EC2 instance store-backed Linux/UNIX AMIs to EBS-backed systems manually.

Important

You can't convert an instance store-backed Windows AMI to an EBS-backed Windows AMI. You must start with a public EBS-backed Windows AMI, modify it to meet your specifications, and then create an image from it. For information, see [Creating Amazon EBS-Backed AMIs Using the Console \(p. 28\)](#).

The following table describes the conversion process.

How to convert a Linux/UNIX Amazon EC2 instance store-backed AMI to an EBS-backed AMI

1	Copy the AMI's root device information to an Amazon EBS volume. For more information, see the related task list in Root Device Storage Usage Scenarios (p. 125)
2	Create a snapshot of that volume. For more information, see Creating an Amazon EBS Snapshot (p. 492) .
3	Register the image with a block device mapping that maps the root device name of your choice to the snapshot you just created. For an example, see Block Device Mapping (p. 522) .

You might find it useful to refer to available blog posts that discuss conversion. The following are two example blogs; AWS, however, takes no responsibility for the completeness or accuracy of the content:

- <http://www.elastician.com/2009/12/creating-ebs-backed-ami-from-s3-backed.html>
- <http://coderslike.us/2009/12/07/amazon-ec2-boot-from-ebs-and-ami-conversion/>

Creating Instance Store-Backed Linux/UNIX AMIs

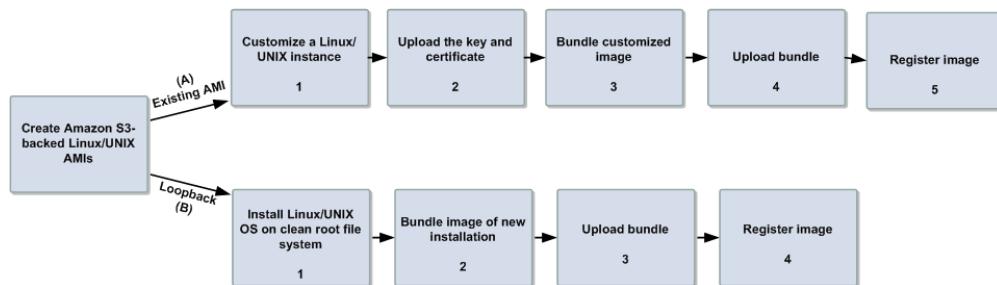
Topics

- [Tools You Need \(p. 35\)](#)
- [Creating an Instance Store-Backed AMI From an Existing AMI \(p. 36\)](#)
- [Creating an Instance Store-Backed AMI From a Loopback \(p. 40\)](#)

For Linux/UNIX systems, you have two common ways to prepare Amazon EC2 instance store-backed AMIs. The easiest method (A) involves launching an existing public AMI and modifying it according to your requirements. For more information, see [Creating an Instance Store-Backed AMI From an Existing AMI \(p. 36\)](#).

Another approach (B) is to build a fresh installation either on a stand-alone machine or on an empty file system mounted by loopback. The process entails building an operating system installation from scratch. After you've built the installation package to your satisfaction, you must bundle it using the AMI tool for bundling volumes and register it using the command line tool for registering images. For information, see [Creating an Instance Store-Backed AMI From a Loopback \(p. 40\)](#).

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



This section discusses the steps for creating AMIs from an existing file and from a loopback, and some basics about the AMI tools.

Tools You Need

Amazon created the Amazon EC2 AMI tools to help you perform specific tasks for Amazon EC2 instance store-backed Linux/UNIX AMIs. You use these AMI tools, which are a set of command line utilities, for bundling and uploading Amazon EC2 instance store-backed Linux/UNIX AMIs. You also use these AMI tools for managing these bundled images. For information about the specific AMI tools, see [AMI Tools Reference](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

When you bundle an Amazon EC2 instance store-backed Linux/UNIX AMI and you start with an instance, you use the AMI tools for bundling and uploading the bundle, and then you use the API tools to register the image. If you are creating an Amazon EC2 instance store-backed AMI from a loopback, you first prepare the instance, then use the AMI tools to bundle before you use the API tools to register the image you created.

If you are starting with an instance of an Amazon public AMI, it might already have the AMI tools installed. Try running the command `ec2-bundle-vol` to check if the instance already has the AMI tools.

If the tools are already installed, you can jump to the section that discusses the bundling process you want to complete:

- [Creating an Instance Store-Backed AMI From an Existing AMI \(p. 36\)](#)
- [Creating an Instance Store-Backed AMI From a Loopback \(p. 40\)](#)

If the tools aren't installed, read on. This section describes installation and usage information when using AMI tools.

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.

For information about installing the AMI tools, see [Amazon EC2 AMI Tools](#).

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
```

View the AMI Tools Documentation

This section describes how to view Linux/UNIX documentation.

To view the manual for each tool

- Append **--manual** to the command that invokes the tool.

```
$ ec2-bundle-image --manual
```

To view help for each tool

- Append **--help** to the command that invokes the tool.

```
$ ec2-bundle-image --help
```

Creating an Instance Store-Backed AMI From 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.

Before You Get Started

1	Before you select an AMI, determine whether the instance types you plan to launch are 32-bit or 64-bit. For more information, see Instance Families and Types (p. 87) .
2	Make sure you are using GNU Tar 1.15 or later.
3	Install the Amazon EC2 API tools. Go to Amazon EC2 API Tools for more information and to download the tools from Amazon S3. Note To ensure you have the latest and most reliable version, we recommend that you install the Amazon EC2 API tools only from Amazon S3.

Tasks to Use an Existing AMI to Create a New AMI

1	Customize an Instance (p. 37)
2	Upload the Key and Certificate (p. 37)
3	Bundle a Customized Image (Requires Root Privileges) (p. 38)
4	Upload a Bundled AMI (p. 39)
5	Register the AMI (p. 40)

Customize an Instance

Customizing an instance involves the following series of steps:

1. Selecting an AMI from available AMIs.
2. Launching an instance from the AMI you selected.
3. Making changes to (thus, *customizing*) the instance, such as altering the Linux configuration, adding software, and configuring web applications.

For more information, see [Launching EC2 Instances \(p. 274\)](#).

After you've launched an instance according to your specifications, proceed to the next steps to create a new AMI using the customized instance.

Upload the Key and Certificate

Your new AMI must be encrypted and signed to ensure that only you and Amazon EC2 can access it. To accomplish this, you must upload your Amazon EC2 private key and X.509 certificate to an instance store directory on your running instance. The private key and the certificate will be used in the AMI bundling process.

The private key and certificate files must not be bundled with the image. To prevent this, create a separate directory for these files. This directory will be specifically excluded from the bundle. In these examples, the private key and certificate files will be stored in the `/tmp/cert` directory. Before you can upload the private key and certificate files, you must grant write permissions to the directory where these files will be uploaded. The following command grants write permission to `/tmp/cert`.

```
$ sudo chmod 777 /tmp/cert
```

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

Copy your Amazon EC2 private key and X.509 certificate to the instance using a secure copy function such as `scp`.

The following shows the syntax to use with the `scp` command.

```
$ scp -i <keypair_name> <private_keyfile> <certificate_file> <username>@<dns_location>:<instance_store_directory>
```

Where,

Parameter	Description
<i>keypair_name</i>	The name of your key pair.
<i>private_keyfile</i>	The file that contains the private key.
<i>certificate_file</i>	The file that contains the certificate.
<i>username</i>	The login name you use to log in to your instance.
<i>dns_location</i>	The DNS location of the instance within Amazon EC2.
<i>instance store directory</i>	The directory where your instance store is mounted.

scp displays the names of the files copied and some performance statistics.

The following is an example of a fully specified **scp** command using the Amazon Linux AMI.

```
$ scp -i gsg-keypair pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-user@ec2-67-202-51-223.compute-1.amazonaws.com:/tmp/cert
pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 717      0.7KB/s  00:00
cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 685      0.7KB/s  00:00
```

Bundle a Customized Image (Requires Root Privileges)

When you have the image that meets your specifications, you need to bundle it for uploading to Amazon S3. The bundling process requires root privileges. You can obtain these privileges with the **sudo su** command.

To bundle a customized image

Use the **ec2-bundle-vol** command. Make sure to exclude the directory where the private key and certificate files are stored with the **-e** option. This option excludes files that may contain sensitive information, such as your AWS credentials.

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

- <*private_keyfile*> is the file that contains the private key
- <*certificate_file*> is the file that contains the certificate
- <*user_id*> is the ID associated with your AWS account. This is your AWS account ID without dashes. It consists of 12 to 15 characters, and it's *not* the same as your access key ID.
- <*cert_location*> is the directory that contains the private key and certificate files, which must be excluded from the bundle.

Note

Be sure to disable SELinux when running **ec2-bundle-vol**.

The following command bundles the local machine root file system.

```
# ec2-bundle-vol -e /tmp/cert -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 111122223333
```

```
Please specify a value for arch [x86_64]:  
Copying / into the image file /tmp/image...  
Excluding:  
/sys  
...  
/tmp/cert  
...  
1+0 records in  
1+0 records out  
1048576 bytes (1.0 MB) copied, 0.00172 s, 610 MB/s  
mke2fs 1.42.3 (14-May-2012)  
Bundling image file...  
Splitting /tmp/image.tar.gz.enc...  
Created image.part.00  
Created image.part.01  
...  
Created image.part.NN  
Generating digests for each part...  
Digests generated.  
Creating bundle manifest...  
ec2-bundle-vol complete.
```

Upload a Bundled AMI

You must upload the bundled AMI to Amazon S3 before Amazon EC2 can access it. This task is necessary when you create Amazon EC2 instance store-backed AMIs from an existing instance or from a loopback. Use the [ec2-upload-bundle](#) command to upload the bundled AMI that you created earlier. Amazon S3 stores data objects in buckets, which are similar to directories. All buckets must have globally unique names. The [ec2-upload-bundle](#) command uploads the bundled AMI to a specified bucket. If the specified bucket exists and belongs to another AWS account, the [ec2-upload-bundle](#) command will fail.

Important

The specified Amazon S3 bucket must exist, and it must have been created in the same region as the instance being uploaded.

To upload the bundled AMI

Use the [ec2-upload-bundle](#) command as follows:

```
$ ec2-upload-bundle -b <your-s3-bucket> -m <manifest_path> -a <access_key> -s <secret_key>
```

- <your-s3-bucket> is the Amazon S3 bucket that the bundle will be uploaded to. You can also upload the bundle to a subfolder of the bucket, such as my-awsbucket/uploaded-images/image-1. If the subfolder does not exist, it will be created.
- <manifest_path> is the full path to the manifest file (for example, /tmp/image.manifest.xml). The manifest file will reside in the destination directory that was specified in the [ec2-bundle-vol](#) command.
- <access_key> is your AWS access key ID.
- <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.

Register the AMI

You must register your image with Amazon EC2, so that Amazon EC2 can locate it and run instances based on it. This task is necessary when you create Amazon EC2 instance store-backed AMIs from an existing file or from a loopback. 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

Use the `ec2-register` command (which is part of the EC2 CLI tools, not the AMI tools) as follows:

```
$ ec2-register <your-s3-bucket>/<path>/image.manifest.xml -n <image_name> -o  
<your_access_key> -W <your_secret_key>
```

Important

The capitalization of the bucket name and path in `<your-s3-bucket>/<path>` must match exactly what was passed in the `ec2-upload-bundle` command.

This command registers the AMI in the default region. To specify a different region, set the `EC2_URL` environment variable, or use the `--region` option with the `ec2-register` command.

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.

Creating an Instance Store-Backed AMI From a Loopback

Creating AMIs through a loopback 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. After you have installed your operating system, you can bundle the resulting image as an AMI with the `ec2-bundle-image` command, which is part of the AMI tools (and not an API action). For more information about the `ec2-bundle-image` command and the AMI tools, go to the [Amazon Elastic Compute Cloud Command Line Reference](#).

Note

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

Before You Get Started

1	Before you select an AMI, determine whether the instance types you plan to launch are 32-bit or 64-bit. For more information, see Instance Families and Types (p. 87) .
2	Make sure you are using GNU Tar 1.15 or later.
3	This topic uses Fedora Core 4. Please make any adjustments for your distribution.

Tasks to Create a New AMI Through a Loopback

1	Install Linux/UNIX and Prepare the System a. Create a File to Host the AMI (p. 41) b. Create a Root File System Inside the File (p. 41) c. Mount the File through Loopback (p. 42) d. Prepare for the Installation (p. 43) e. Install the Operating System (p. 44) f. Configure the Operating System (p. 45)
2	Bundle the Loopback File Image (p. 46)
3	Upload a Bundled AMI (p. 47)
4	Register the AMI (p. 48)

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/UNIX installation requires about 700 MB, 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 example creates a 1 GB file (1024*1 MB).

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

Create a Root File System Inside the File

The `mksfs` utility has several variations that can create a file system inside the image file you are creating. Typical Linux/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 example 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.
```

Mount the File through Loopback

The loopback module enables 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. Enter the following command to 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
```

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

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

```
[fedora]
name=Fedora $releasever - $basearch
failovermethod=priority
#baseurl=http://download.fedoraproject.org/pub/fedora/linux/releases/$releas
ever/Everything/$basearch/os/
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=fedora-$releas
ever&arch=$basearch
enabled=1
#metadata_expire=7d
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$basearch

[updates]
name=Fedora $releasever - $basearch - Updates failovermethod=priority
#baseurl=http://download.fedoraproject.org/pub/fedora/linux/updates/$releas
ever/$basearch/
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=updates-released-
f$releasever&arch=$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$basearch
```

This step ensures that all the required basic packages and utilities are installed. You can locate this file anywhere on your main file system (not on your loopback file system) and is used only 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/MAKDEV -d /mnt/ec2-fs/dev -x console
MAKDEV: mkdir: File exists
MAKDEV: mkdir: File exists
MAKDEV: mkdir: File exists
# /sbin/MAKDEV -d /mnt/ec2-fs/dev -x null
MAKDEV: mkdir: File exists
MAKDEV: mkdir: File exists
MAKDEV: mkdir: File exists
# /sbin/MAKDEV -d /mnt/ec2-fs/dev -x zero
MAKDEV: mkdir: File exists
MAKDEV: mkdir: File exists
MAKDEV: 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
```

Install the Operating System

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

To install the operating system

- Enter the following command:

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

The *<yum_configuration_file>* is the name of the YUM configuration file and *<image_mountpoint>* is the mount location.

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

Example

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

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

Configure the Operating System

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

To configure the operating system

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

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

Note

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

2. Verify that the following line appears in the `/mnt/ec2-fs/etc/sysconfig/network` file so that networking starts:

```
NETWORKING=yes
```

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

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

Note

The `/dev/sda2` and `/dev/sda3` storage locations only apply to small instances. For more information on instance storage, see the section called “[Instance Store](#)” (p. 513).

4. Allocate appropriate system run levels so that all your required services start at system startup. For example, to enable a service on multiuser and networked run levels, use 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. Enter the following commands to umount the image:

```
# 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
```

Bundle the Loopback File Image

To bundle the loopback file image

- Enter the following command:

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

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

Note

The user ID is your AWS account ID without dashes. It consists of 12 to 15 characters, and it's *not* the same as your Access Key ID.

Example

The [ec2-bundle-image](#) command bundles an image created in a loopback file.

```
# ec2-bundle-image -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c cert-HKZYK
TAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 111122223333 -i image.img -d bundled/ -p
fred -r x86_64
Please specify a value for arch [i386]:
Bundling image file...
Splitting bundled/fred.gz.crypt...
Created fred.part.00
Created fred.part.01
Created fred.part.02
Created fred.part.03
Created fred.part.04
Created fred.part.05
Created fred.part.06
Created fred.part.07
Created fred.part.08
Created fred.part.09
Created fred.part.10
Created fred.part.11
Created fred.part.12
Created fred.part.13
Created fred.part.14
Generating digests for each part...
Digests generated.
Creating bundle manifest...
ec2-bundle-image complete.
```

Upload a Bundled AMI

You must upload the bundled AMI to Amazon S3 before Amazon EC2 can access it. This task is necessary when you create Amazon EC2 instance store-backed AMIs from an existing instance or from a loopback. Use the [ec2-upload-bundle](#) command to upload the bundled AMI that you created earlier. Amazon S3 stores data objects in buckets, which are similar to directories. All buckets must have globally unique names. The [ec2-upload-bundle](#) command uploads the bundled AMI to a specified bucket. If the specified bucket exists and belongs to another AWS account, the [ec2-upload-bundle](#) command will fail.

Important

The specified Amazon S3 bucket must exist, and it must have been created in the same region as the instance being uploaded.

To upload the bundled AMI

Use the [ec2-upload-bundle](#) command as follows:

```
$ ec2-upload-bundle -b <your-s3-bucket> -m <manifest_path> -a <access_key> -s
<secret_key>
```

- <your-s3-bucket> is the Amazon S3 bucket that the bundle will be uploaded to. You can also upload the bundle to a subfolder of the bucket, such as my-awsbucket/uploaded-images/image-1. If the subfolder does not exist, it will be created.
- <manifest_path> is the full path to the manifest file (for example, /tmp/image.manifest.xml). The manifest file will reside in the destination directory that was specified in the [ec2-bundle-vol](#) command.
- <access_key> is your AWS access key ID.

- <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.

Register the AMI

You must register your image with Amazon EC2, so that Amazon EC2 can locate it and run instances based on it. This task is necessary when you create Amazon EC2 instance store-backed AMIs from an existing file or from a loopback. 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

Use the `ec2-register` command (which is part of the EC2 CLI tools, not the AMI tools) as follows:

```
$ ec2-register <your-s3-bucket>/<path>/image.manifest.xml -n <image_name> -O  
<your_access_key> -W <your_secret_key>
```

Important

The capitalization of the bucket name and path in `<your-s3-bucket>/<path>` must match exactly what was passed in the `ec2-upload-bundle` command.

This command registers the AMI in the default region. To specify a different region, set the `EC2_URL` environment variable, or use the `--region` option with the `ec2-register` command.

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.

Creating and Launching an AMI from a Snapshot

If you have a snapshot of the root device volume of an instance, you can terminate that instance and then later launch a new instance from the snapshot. You must first register the snapshot, then create and launch the resulting AMI, as explained in [Creating and Launching an AMI from a Snapshot \(p. 290\)](#).

Creating Paid AMIs

Topics

- [How is AWS Marketplace different than Amazon DevPay? \(p. 49\)](#)
- [Amazon DevPay \(p. 49\)](#)
- [Product Registration \(p. 53\)](#)
- [Associating a Product Code with an AMI \(p. 53\)](#)
- [Sharing Your Paid AMI \(p. 54\)](#)
- [Confirming an Instance is Running with a Product Code \(p. 55\)](#)
- [Getting the Product Code from Within an Instance \(p. 55\)](#)
- [Supported AMIs \(p. 56\)](#)

A *paid AMI* is an AMI that you sell to other Amazon EC2 users. These users pay you according to the price you set. AWS Marketplace and Amazon DevPay provide capabilities to create and distribute paid AMIs. This section gives an introduction to the AWS Marketplace and Amazon DevPay.

The AWS Marketplace is an online store that helps customers find, compare, and immediately start using the software they need to build products and run their businesses. For an overview of the AWS Marketplace, go to <https://aws.amazon.com/marketplace/help>. You can also visit the [AWS Marketplace](#).

Sellers interested in submitting AMIs to AWS Marketplace should consult the [AWS Marketplace Seller's Guide](#).

AWS Marketplace uses the same approach to validate product codes, see [Confirming an Instance is Running with a Product Code \(p. 55\)](#) and [Getting the Product Code from Within an Instance \(p. 55\)](#). However, other details of creating, submitting AMIs to the AWS Marketplace are different, and are discussed in more detail in the [AWS Marketplace Seller's Guide](#).

Important

Amazon DevPay does not support Amazon EBS-backed AMIs. All paid AMIs must be backed by Amazon instance store.

How is AWS Marketplace different than Amazon DevPay?

There are substantial differences between AWS Marketplace and Amazon DevPay. Both help customers buy software that runs on AWS, but AWS Marketplace offers a more comprehensive experience. For software buyers the key differences are:

- AWS Marketplace offers a more Amazon.com-like shopping experience, simplifying discovery of available software.
- AWS Marketplace products work with other AWS features, such as Amazon VPC, and can be run on Reserved and Spot instances in addition to normal On Demand Instances.
- AWS Marketplace supports EBS-backed software, where Amazon DevPay does not.

Amazon DevPay

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.

Important

Amazon DevPay does not support Amazon EBS-backed AMIs. All paid AMIs must be backed by Amazon instance store.

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. Amazon DevPay offers the following:

- You can charge customers for your Amazon EC2 instance store-backed 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 Amazon EC2 instance store-backed 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, Amazon DevPay turns off their access to your AMI for you.
- Amazon Payments handles payment processing.



Basic Amazon DevPay Flow

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

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

Summary of How Paid AMIs Work with Amazon DevPay

With a paid AMI using Amazon DevPay, your customers:

- Must be signed up to use Amazon EC2 themselves
- Buy your paid Amazon EC2 instance store-backed 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

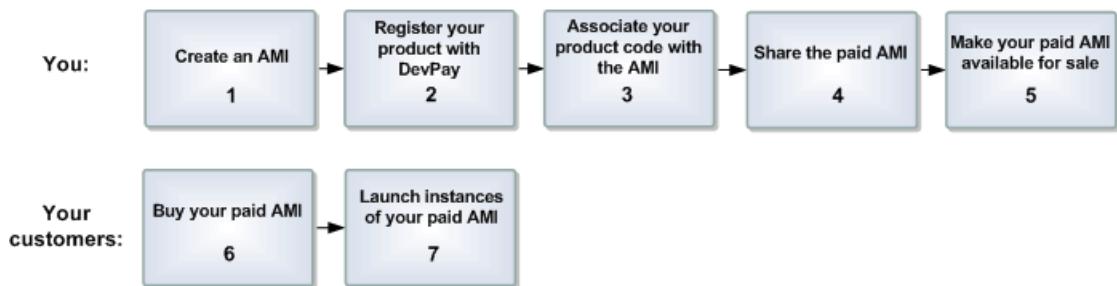
The discounts you get with Amazon EC2 Reserved Instances don't apply to Amazon DevPay products. That is, if you purchase Reserved Instances, you don't get the lower price associated with them when your customers launch your paid or supported AMIs. Also, if your customers purchase Reserved Instances, 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 Instances \(p. 212\)](#).

You can also use Amazon EC2 and Amazon DevPay together with a *supported AMI*. For more information about supported AMIs, see [Supported AMIs \(p. 56\)](#).

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 Amazon EC2 instance store-backed AMI as described elsewhere in this guide.
2	<p>You register a product with Amazon DevPay.</p> <p>For more information, see Product Registration (p. 53). As part of this process, you provide a product description, product pricing, etc. This registration process creates a product code for the product and a URL where customers can sign up to use the product (called the <i>purchase URL</i>).</p> <p>Note</p> <p>You cannot register an AMI that already has an AWS Marketplace code associated with it.</p>
3	<p>You use an Amazon EC2 command or API call to associate the product code with your Amazon EC2 instance store-backed AMI.</p> <p>For more information, see Associating a Product Code with an AMI (p. 53). This makes the AMI a paid AMI.</p>
4	<p>You use an Amazon EC2 command or API call to share the Amazon EC2 instance store-backed AMI with select customers or the public.</p> <p>For more information, see Sharing Your Paid AMI (p. 54).</p> <p>Note</p> <p>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).</p>
5	<p>You make your paid AMI available for sale.</p> <p>To do this, you make the aforementioned purchase URL available. You can advertise your paid AMI in the https://forums.aws.amazon.com/ <i>Solutions Catalog</i> on the AWS Developer Connection site.</p>
6	<p>Customers use the purchase URL you provide to sign up for and purchase your product.</p> <p>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).</p>
7	<p>Customers then launch an Amazon EC2 instance specifying the AMI ID.</p> <p>Because you associated the shared AMI with the product code, the customers are charged at the rate you set. For more information, see Paid AMIs (p. 26).</p>

Note

You can associate your Amazon DevPay product code with more than one Amazon EC2 instance store-backed 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 Amazon 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 Amazon 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 Amazon EC2 instance store-backed AMI and sell the AMI as a Amazon DevPay product. There's another scenario for using Amazon 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 Amazon EC2 instance store-backed AMIs. For more information, see [Supported AMIs \(p. 56\)](#).

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 [Creating Your Own AMIs \(p. 30\)](#)). 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, uses modified versions of the AWS tools, or bundles the AMI outside of Amazon EC2, 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.

For increased security, we recommend that you configure your application to check the instance metadata to verify that the product code is intact. For more information about retrieving metadata, see [Instance Metadata and User Data \(p. 365\)](#).

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 [Confirming an Instance is Running with a Product Code \(p. 55\)](#).

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 [Getting the Product Code from Within an Instance \(p. 55\)](#).

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 [Creating Your Own AMIs \(p. 30\)](#)). 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 Amazon 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 Amazon EC2 instance store-backed AMI that you want to sell or multiple Amazon EC2 instance store-backed 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, which contains a product code and type
- Product token
- Purchase URL

You need the product code and purchase URL to integrate your product with Amazon DevPay as described in [Summary of How Paid AMIs Work with Amazon DevPay \(p. 50\)](#) and [Supported AMIs \(p. 56\)](#). You need the product token if you're going to set up your system to later verify whether a customer is still subscribed to your product. For more information, refer to the *Amazon DevPay Developer Guide*.

Note

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

Associating a Product Code with an AMI

You must be the owner of an Amazon EC2 instance store-backed 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      [devpay: 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      [devpay: 774F4FF8]
```

Sharing Your Paid AMI

After you associate the product code with the Amazon EC2 instance store-backed 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
```

Confirming 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. This applies to both Amazon DevPay AMIs and AWS Marketplace AMIs.

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 (Amazon DevPay or AWS Marketplace)

- Enter the following command:

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

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

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

Example

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

```
PROMPT> ec2-confirm-product-instance 6883959E -i i-10a64379
          6883959E i-10a64379 true 495219933132
```

Getting the Product Code from Within an Instance

A running Amazon EC2 instance can determine if it has an Amazon DevPay or AWS Marketplace product code. The instance retrieves the product code similarly to how it retrieves other metadata. For more information about retrieving metadata, see [Instance Metadata and User Data \(p. 365\)](#).

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 Amazon DevPay product. For more information, see [Reserved Instances \(p. 212\)](#).

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



Supported AMI Process

1	<p>You register a product with Amazon DevPay.</p> <p>For more information, see Product Registration (p. 53). As part of this process, you provide a product description, product pricing, etc. This registration process creates a product code for the product and a URL where customers can sign up to use the product (called the <i>purchase URL</i>).</p>
2	<p>You make your product available for sale.</p>
3	<p>Customers use the purchase URL to sign up for and purchase your product.</p> <p>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).</p>
4	<p>Customers then use an Amazon EC2 command or API call to associate the product code with their Amazon EC2 instance store-backed AMIs.</p> <p>For more information, see Associating a Product Code with an AMI (p. 53).</p>
5	<p>Customers then launch one or more instances of the AMIs.</p> <p>Because the customers associated their AMIs with the product code, they are charged at the rate you set.</p>

Note

Amazon EC2 prevents your customers (but not you as the product code owner) from associating your product code with AMI types for which the product isn't configured. 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 for which your product isn't configured. 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 Amazon 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 [Confirming an Instance is Running with a Product Code \(p. 55\)](#).

Sharing AMIs Safely

Topics

- [Protecting a Shared AMI \(Linux/UNIX\) \(p. 58\)](#)

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

For additional information about sharing AMIs safely, go to the following articles on the AWS Developer Resources website.

- [How To Share and Use Public AMIs in A Secure Manner](#)
- [Public AMI Publishing: Hardening and Clean-up Requirements](#)

For information on building shared AMIs, see [Protecting a Shared AMI \(Linux/UNIX\) \(p. 58\)](#). For information on sharing AMIs, see [Sharing AMIs \(p. 62\)](#).

Protecting a Shared AMI (Linux/UNIX)

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

To build a shared AMI, follow these guidelines:

Shared AMI Guidelines

1	Update the AMI Tools at Boot Time (p. 58)
2	Disable Password-Based Logins for Root (p. 59)
3	Restrict Access to the Root Account (p. 60)
4	Install Public Key Credentials (p. 60)
5	Disabling sshd DNS Checks (optional) (p. 61)
6	Identify Yourself (p. 61)
7	Protect Yourself (p. 61)
8	Protect Paid AMIs (p. 62)

Update the AMI Tools at Boot Time

For AMIs backed by instance store, 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 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.

The preceding procedure applies to RPM-based 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 restrict access to the root account and 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/firstrun file is bundled with the image. If the file was not created, the root password will never be randomized and is set to the default.

Restrict Access to the Root Account

When you work with shared AMIs, it is a known best practice to have a secure environment; one of the elements associated with a secure environment is ensuring the root password is not empty. To do this, log into your running instance and issue the following command to lock the root password:

```
sudo passwd -l root
```

Note

This does not impact the use of sudo.

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

Important

If you forget to remove the existing SSH host key pairs from your public AMI, our routine auditing process notifies you and all customers running instances of your AMI of the potential security risk. After a short grace period, we mark the AMI private.

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 or ImportKeyPair) is made available to the instance through an HTTP query against the instance metadata.

To log in 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 can launch instances of your AMI with a key pair and log in without requiring a root password.

```
if [ ! -d /root/.ssh ] ; then
    mkdir -p /root/.ssh
    chmod 700 /root/.ssh
fi
```

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

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

Note

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

Disabling sshd DNS Checks (optional)

Disabling sshd DNS checks slightly weakens your sshd security. However, if DNS resolution fails, SSH logins 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, because each AMI is represented by an account ID.

We recommend that you post a description of your AMI, and the AMI ID, in the [Amazon EC2 forum](#). This provides a convenient central location for users who are interested in trying new shared AMIs. You can also post the AMI to the [Amazon Machine Images \(AMIs\)](#) page.

Protect Yourself

The previous sections described how to make your shared AMIs safe, secure, and usable 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 on 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 you to share your AMIs with other AWS accounts. This section describes how to share AMIs using the Amazon EC2 management console and the command line tools.

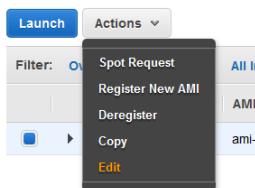
Note

Before proceeding, make sure to read the security considerations of sharing AMIs in the [Protecting a Shared AMI \(Linux/UNIX\) \(p. 58\)](#) section.

Sharing an AMI Using the Management Console

To share an AMI using the management console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **AMIs**.
3. Select your AMI in the list, then click **Actions** and select **Edit**.



4. Expand the **Permissions** area and ensure that the **Private** radio button is selected.
5. Click **Add Permission**, and type the AWS Account Number of the user with whom you want to share the AMI in the **AWS Account Number** field. Click **Save** (the checkmark button).

▼ Permissions

This image is currently: Public Private

Add "create volume" permissions to the following associated snapshots when creating permissions:
• snap-67dc9220

AWS Account Number	Actions
1234567891011	<input checked="" type="checkbox"/> <input type="checkbox"/>

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

Note

You do not need to share the Amazon EBS snapshots that an AMI references in order to share the AMI. Only the AMI itself needs to be shared; the system automatically provides the instance access to the referenced Amazon EBS snapshots for the launch.

Accessing Shared AMIs Using the Amazon EC2 Console

If another AWS user has shared an AMI with you, you can access it using the Amazon EC2 console.

To access a shared AMI using the console

1. In the navigation pane, click **AMIs**.
2. In the **Filter** list, select **Private Images**. All AMIs that have been shared with you are listed.

Using Bookmarks

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

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

To create a bookmark for your AMI

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

```
https://console.aws.amazon.com/ec2/home?region=<region>#launchAmi=<ami_id>
```

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

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

2. Distribute the link to users who want to use your AMI.
3. To use a bookmark, click the link or copy and paste it into your browser. The Launch Wizard opens, and the **CHOOSE AN AMI** page is pre-populated with the AMI's details.

Sharing an AMI Using the Command Line Tools

This section explains how to share an AMI using the command line tools.

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

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

Making an AMI Public

This section explains how to make an AMI public.

Note

Images with an AWS Marketplace product code cannot be made 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 does 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 example 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
```

Sharing an AMI with Specific AWS Accounts

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

To grant explicit launch permissions

- Enter the following command:

```
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 account ID, without hyphens.

To remove launch permissions for an account

- 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 account ID, without hyphens.

To remove all launch permissions

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

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

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

Note

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

Example

The following example grants launch permissions to the AWS account with ID 111122223333 for the ami-2bb65342 AMI:

```
PROMPT> ec2-modify-image-attribute ami-2bb65342 -l -a 111122223333
launchPermission      ami-2bb65342      ADD      userId 111122223333
```

The following example removes launch permissions from the AWS account with ID 111122223333 for the ami-2bb65342 AMI:

```
PROMPT> ec2-modify-image-attribute ami-2bb65342 -l -r 111122223333
launchPermission      ami-2bb65342      REMOVE    userId 111122223333
```

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
```

Publishing Shared AMIs

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

To publish your AMI

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

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

    <li>Published by [Publisher] (<a href="http://www.example.com">[ht
tp://www.example.com]</a>).<br />
        </li>
    <li>[Key Features]<br />
        </li>
    <li>[Description]</li>
    <li>This image contains the following daemons / services:
        <ul>
```

```
<li>[Daemon 1]</li>
<li>[Daemon 2]</li>
</ul>
</li>
</ul>
<h2><strong>What's New?</strong></h2>The following changes were made on [Date].<br />
<ul>

    <li>[Release Notes 1]</li>
</ul>
<span style="font-size: x-small; font-family: courier new,courier">&nbsp;&nbsp;&nbsp;&nbsp;-&nbsp;[Note 1]</span><br />
<span style="font-size: x-small; font-family: courier new,courier">&nbsp;&nbsp;&nbsp;&nbsp;-&nbsp;[Note 2]</span><br />
<span style="font-size: x-small; font-family: courier new,courier">&nbsp;&nbsp;&nbsp;&nbsp;-&nbsp;[Note 3]</span><br />
<!--<ul>-->
```

Copying AMIs

With EC2 AMI copy, you can easily copy the Amazon Machine Images (AMIs) you own to other AWS regions and scale your applications to take advantage of AWS's geographically diverse regions.

EC2 AMI copy provides the following benefits:

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

EC2 AMI copy enables you to copy an AMI to as many regions as you like from the AWS Management Console, the Amazon EC2 CLI, or the Amazon EC2 API. EC2 AMI copy is available for Amazon EBS-backed AMIs as well as instance-store-backed AMIs, and is operating-system-agnostic.

Important

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

Each copy of an AMI results in a new AMI with its own unique AMI ID. Any changes made to the source AMI during or after a copy are not propagated to the new AMI as part of the AMI copy process; you must recopy the AMI to the destination regions to copy the changes made to the source AMI.

Note

Permissions and user-defined tags applied to the source AMI are not copied to the new AMIs as part of the AMI copy process. After the copy is complete, you may apply any permissions and user-defined tags to the new AMIs.

Copying an Amazon EC2 AMI

You can copy an EC2-Classic, default VPC AMI using the AWS Management Console, the EC2 CLI tools, or the EC2 API.

AWS Management Console

To copy an AMI using the AWS Management Console

You can create a copy of an Amazon EC2 AMI using the console. When you copy an AMI using the console, a copy of the source AMI is "pushed" to the destination region.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **AMIs**.
3. Select the AMI to copy, click **Actions** and select **Copy**.
4. In the AMI Copy page, choose the following:
 - **Destination region:** Select the region to which you want to copy the AMI.
 - **Name:** Type a name for the new AMI.
 - **Description:** By default, the description includes information about the source AMI so that you can identify a copy from the original. You can change this description as necessary.
5. Click **Copy**.
6. The Copy AMI confirmation page appears, notifying you that the copy operation has been initiated. The new AMI ID is provided. Click **Done**.

To view the progress of the copy operation later, switch the Amazon EC2 console to the destination region, and then refresh the AMI pane. The copy operation is complete when the new AMI's status displays available.

Command Line Tools

To copy an AMI using the CLI

You can copy an AMI using the CLI by completing the instructions in the [ec2-copy-image topic](#) in the *Amazon Elastic Compute Cloud Command Line Reference*. This action initiates the copy operation and registers the new AMI in the destination region.

API

To copy an AMI using the API

You can create a copy of an Amazon EC2 AMI using the API by completing the steps in the [CopyImage topic](#) in the *Amazon Elastic Compute Cloud API Reference*. This action connects to the destination region and "pulls" copies of the source AMI to that destination region.

Stopping the AMI copy operation

You can stop the copy operation of an EC2-Classic, default VPC AMI using the AWS Management Console, the CLI, or the API.

AWS Management Console

To stop the AMI copy operation using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Select the destination region from the region selector.
3. In the navigation pane, click **AMIs**.
4. Select the AMI you want to stop copying, click **Actions** and select **Deregister**.
5. When asked for confirmation, click **Continue**.

Command Line Tools

To stop the AMI copy operation using the CLI

You can stop the AMI copy operation using the CLI by completing the instructions in the [ec2-deregister topic](#) in the *Amazon Elastic Compute Cloud Command Line Reference*. This action stops the copy operation and deregisters the new AMI in the destination region.

API

To stop the AMI copy operation using the API

You can stop the AMI copy operation using the API by completing the steps in the [DeregisterImage topic](#) in the *Amazon Elastic Compute Cloud Command Line Reference*. This action stops the copy operation and deregisters the new AMI in the destination region.

Frequently Asked Questions

This section addresses frequently asked questions about EC2 AMI copy.

Q: What kind of AMIs can I copy with EC2 AMI copy?

You can only copy AMIs that you own. You cannot copy Amazon or community AMIs.

You can only copy AMIs that you own. You cannot copy Amazon, community, or Marketplace AMIs.

Q: Who can copy AMIs?

Any AWS customer who has access to the AWS Management Console or APIs and CLIs can copy an AMI. IAM users who are authorized users or owners of an AMI can also copy AMIs.

Q: Will permissions and tags from the source AMI be copied to the destination AMI?

No. Permissions and user-defined tags are not copied. Permissions and user-defined tags that have been granted to the source AMI must be created again on the destination AMI after the copy is complete.

- For more information about how to add permissions, go to [Setting Launch Permissions on an Amazon Machine Image](#) in the *AWS Toolkit for Visual Studio User Guide*.
- For more information about how to add user-defined tags, see [Tagging Your Amazon EC2 Resources \(p. 540\)](#).

Q: How do I know when the copy operation is complete?

After the new AMI in the destination region displays a status of *available*, the copy operation is complete and the AMI is ready to be used.

Q: After I have copied an AMIs, will subsequent updates to the source AMI be propagated to my copied AMIs in the destination region?

No. EC2 AMI copy results in two distinct AMIs, the original AMI in the source region and the new AMI in the destination region. Any subsequent changes to the source AMI are independent of the copied AMIs in the destination region, and vice versa. If you make changes to the source AMIs and want those changes to be reflected in the copied AMIs in the destination region, you must recopy the AMI to the destination regions.

Q: After an AMI has been copied, can I make changes to it independently of the original (source) AMI?

Yes. The new AMI is fully independent of the source AMI, and it can be modified without affecting the source AMI. The reverse is also true: the source AMI can be modified without affecting the new AMI.

Q: Are new AMIs linked back to the original (source) AMI?

No. There is no link between the source AMI and the new AMI.

Q: Which AKI and ARI will the new AMI use?

EC2 AMI copy tries to find matching AKIs and ARIs for the new AMI in the selected destination region. If no suitable AKI or ARI can be found, then the AMI is not copied. If you are using the AKIs and ARIs recommended by Amazon, the copy operation registers the AMI with the appropriate AKI and ARI in the destination region.

Q: What does the error message "Failed to find matching AKI/ARI" mean?

If you get an error message "Failed to find matching AKI/ARI", that means that the destination region does not contain an equivalent AKI/ARI to those specified in the AMI. If your AMI is using a PV-GRUB AKI, then you can update the AMI to leverage the latest version of PV-GRUB. For more information on PV-GRUB and AKIs, see [Enabling Your Own Linux Kernels](#).

Q: Are copy requests submitted to the source or the destination region endpoint?

Calls to request a copy of an AMI from the CLI or API are submitted to and initiated from the destination region endpoint.

Q: What types of AMIs and operating systems does EC2 AMI copy support?

EC2 AMI copy works with all AMIs and operating systems supported by Amazon EC2.

How much does EC2 AMI copy cost?

There are no charges for EC2 AMI copy. However, standard storage and data transfer rates apply.

Q: Will EC2 AMI copy cost more for Windows AMIs?

No. Copying Windows AMIs is free, just like copying Linux AMIs. Currently, differential rates only apply to running instances and reflect the increased cost of running a Windows license. A Windows AMI that is the same size as a Linux AMI and that uses the same backing store (EBS or instance) will result in the same storage and bandwidth usage charges after the copy is complete.

Amazon Linux AMIs

The Amazon Linux AMI is a supported and maintained Linux image provided by Amazon Web Services (AWS) for use on Amazon Elastic Compute Cloud (Amazon EC2). It is designed to provide a stable, secure, and high-performance execution environment for applications running on Amazon EC2. It also includes packages that enable easy integration with AWS, including launch configuration tools and many popular AWS libraries and tools. Amazon Web Services provides ongoing security and maintenance updates to all instances running the Amazon Linux AMI. The Amazon Linux AMI is provided at no additional charge to Amazon EC2 users.

Topics

- [Finding the Amazon Linux AMI \(p. 71\)](#)
- [Launching and Connecting to an Amazon Linux Instance \(p. 71\)](#)
- [Identifying Amazon Linux AMI Images \(p. 72\)](#)
- [Included AWS Command Line Tools \(p. 72\)](#)
- [cloud-init \(p. 73\)](#)
- [Repository Configuration \(p. 74\)](#)
- [Adding Packages \(p. 75\)](#)
- [Accessing Source Packages for Reference \(p. 75\)](#)
- [Developing Applications \(p. 76\)](#)
- [Instance Store Access \(p. 76\)](#)
- [Product Life Cycle \(p. 76\)](#)
- [Security Updates \(p. 76\)](#)
- [Support \(p. 77\)](#)
- [Enabling Your Own Linux Kernels \(p. 77\)](#)

Finding the Amazon Linux AMI

For a list of the latest Amazon Linux AMIs, see [Amazon Linux AMIs](#).

Launching and Connecting to an Amazon Linux Instance

After locating your desired AMI, note the AMI ID. You can use the AMI ID to launch and then connect to your instance.

The Amazon Linux AMI does not allow remote root SSH by default. Also, password authentication is disabled to prevent brute-force password attacks. To enable SSH logins to a running Amazon Linux AMI, you must provide your key pair to the instance at launch. You must also set the security group used to launch your instance to allow SSH access. By default, the only account that can log in remotely via SSH is ec2-user. The ec2-user has sudo privileges. If you want to enable remote root log in, please be aware that it is less secure than relying on key pairs and a secondary user.

For information on launching and using your Amazon Linux instance, see [Launching EC2 Instances \(p. 274\)](#). For information on connecting to your Amazon Linux instance, see [Connecting to Linux/UNIX Instances from Linux/UNIX using SSH \(p. 297\)](#).

Identifying Amazon Linux AMI Images

Each image contains a unique `/etc/image-id` that identifies the AMI. This file contains information about the image.

Following is an example of the `/etc/image-id` file:

```
$ cat /etc/image-id
image_name="amzn-ami-pv"
image_version="2013.03"
image_arch="x86_64"
image_file="amzn-ami-pv-2013.03.0.x86_64.ext4"
image_stamp="7c35-50b2"
image_date="20130322053523"
recipe_name="amzn ami"
recipe_id="240049bc-8990-fb91-ccad-aeb9-4dfe-eaac-44035533"
```

The `image_name`, `image_version`, and `image_arch` items come from the build recipe that Amazon used to construct the image. The `image_stamp` is simply a unique random hex value generated during image creation. The `image_date` item is in `YYYYMMDDhhmmss` format, and is the UTC time of image creation. The `recipe_name` and `recipe_id` refer to the name and ID of the build recipe Amazon used to construct the image, which identifies the current running version of the Amazon Linux AMI. This file will not change as you install updates from the yum repository.

Amazon Linux AMIs contain a `/etc/system-release` file that specifies the current release that is installed. This file is updated through yum and is part of the system-release rpm.

Following is an example of a `/etc/system-release` file:

```
# cat /etc/system-release
Amazon Linux AMI release 2013.03
```

An Amazon Linux AMI also contains a machine readable version of the `/etc/system-release` file found in `/etc/system-release-cpe` and follows the CPE specification from MITRE ([CPE](#)).

Included AWS Command Line Tools

The following popular command line tools for AWS integration and usage have been included in the Amazon Linux AMI or in the repositories:

- `aws-amitools-ec2`
- `aws-apitools-as`
- `aws-apitools-cfn`
- `aws-apitools-common`
- `aws-apitools-ec2`
- `aws-apitools-elb`
- `aws-apitools-iam`
- `aws-apitools-mon`
- `aws-apitools-rds`
- `aws-cfn-bootstrap`
- `aws-cli`
- `aws-scripts-ses`

To simplify the configuration of these tools, a simple script has been included to prepare AWS_CREDENTIAL_FILE, JAVA_HOME, AWS_PATH, PATH, and product-specific environment variables after a credential file has been installed.

Also, to allow the installation of multiple versions of the API and AMI tools, we have placed symlinks to the desired versions of these tools in `/opt/aws`, as described here:

- `/opt/aws/bin`—Symlink farm to `/bin` directories in each of the installed tools directories.
- `/opt/aws/{apitools|amitools}`—Products are installed in directories of the form [name]-version and symlink [name] attached to the most recently installed version.
- `/opt/aws/{apitools|amitools}/{name}/environment.sh`—Used by `/etc/profile.d/aws-apitools-common.sh` to set product-specific environment variables (EC2_HOME, etc.).

cloud-init

The cloud-init package is an open source application built by Canonical that is used to bootstrap Linux images in a cloud computing environment, such as Amazon EC2. The Amazon Linux AMI contains a customized version of cloud-init. It enables you to specify actions that should happen to your instance at boot time. You can pass desired actions to cloud-init through the user data fields when launching an instance. This means you can use common AMIs for many use cases and configure them dynamically at startup. The Amazon Linux AMI also uses cloud-init to perform initial configuration of the ec2-user account.

For more information about cloud-init, see <https://help.ubuntu.com/community/CloudInit>.

The Amazon Linux AMIs use the following cloud-init actions (configurable in `/etc/sysconfig/cloudinit`):

- action: INIT (always runs)
 - Setting a default locale.
 - Setting the hostname.
 - Parsing and handling user data.
- action: CONFIG_SSH
 - Generating host private SSHkeys.
 - Adding user's public SSHkeys to `.ssh/authorized_keys` for easy login and administration.
- action: PACKAGE_SETUP
 - Preparing yum repo.
 - Handles package actions defined in user data.
- action: RUNCMD
 - Runs a shell command.
- action: RUN_USER_SCRIPTS
 - Executes user scripts found in user data.
- action: CONFIG_MOUNTS
 - Mounts ephemeral drives.
- action: CONFIG_LOCALE
 - Sets the locale in the locale config file according to user data.

Supported User-Data Formats

The cloud-init package supports user-data handling of a variety of formats:

- Gzip

- If user-data is gzip compressed, cloud-init will decompress the data and handle as appropriate.
- MIME multipart
 - Using a MIMI multipart file, you can specify more than one type of data. For example, you could specify both a user-data script and a cloud-config type. Each part of the multipart file can be handled by cloud-init if it is one of the supported formats.
- Base64 decoding
 - If user-data is base64 encoded, cloud-init determines if it can understand the decoded data as one of the supported types. If it understands the decoded data, it will decode the data and handle as appropriate. If not, it returns the base64 data intact.
- User-Data script
 - Begins with "#!" or "Content-Type: text/x-shellscrip".
 - The script will be executed by "/etc/init.d/cloud-init-user-scripts" level during first boot. This occurs late in the boot process (after the initial configuration actions were performed).
- Include file
 - Begins with "#include" or "Content-Type: text/x-include-url".
 - This content is an include file. The file contains a list of URLs, one per line. Each of the URLs will be read, and their content will be passed through this same set of rules. The content read from the URL can be gzipped, MIME-multi-part, or plain text.
- Cloud Config Data
 - Begins with "#cloud-config" or "Content-Type: text/cloud-config".
 - This content is cloud-config data. See the examples for a commented example of supported config formats.
- Cloud Boothook
 - Begins with "#cloud-boothook" or "Content-Type: text/cloud-boothook".
 - This content is boothook data. It is stored in a file under /var/lib/cloud and then executed immediately.
 - This is the earliest "hook" available. Note that there is no mechanism provided for running it only once. The boothook must take care of this itself. It is provided with the instance ID in the environment variable INSTANCE_ID. Use this variable to provide a once-per-instance set of boothook data.

Repository Configuration

Beginning with the 2011.09 release of the Amazon Linux AMI, Amazon Linux AMIs are treated as snapshots in time, with a repository and update structure that always gives you the latest packages when you run `yum update -y`.

The repository structure is configured to deliver a continuous flow of updates that allow you to roll from one version of the Amazon Linux AMI to the next. For example, with the release of the 2013.03 Amazon Linux AMI, if you launch older versions of the Amazon Linux AMI (such as 2012.09 or earlier) and run `yum update -y`, you will still end up with the latest packages.

You can disable rolling updates for the Amazon Linux AMI by enabling the *lock-on-launch* feature. The lock-on-launch feature locks your newly launched instance to receive updates only from the specified release of the AMI. For example, you can launch a 2012.03 AMI and have it receive only the updates that were released prior to the 2013.03 Amazon Linux AMI, until you are ready to migrate to the 2013.03 AMI. To enable lock-on-launch in new instances, launch a 2013.03 Amazon Linux AMI with the following user data passed to cloud-init, using either the EC2 console or the `ec2-run-instances` command with the `-f` flag.

```
#cloud-config
repo_releasever:2013.03
```

To lock existing instances to their current AMI release version

1. Edit `/etc/yum.conf`.
2. Comment out `releasever=latest`.
3. Run `yum clean all` to clear the cache.

Adding Packages

The Amazon Linux AMI is designed to be used with online package repositories hosted in each Amazon EC2 region. These repositories provide ongoing updates to packages in the Amazon Linux AMI, as well as access to hundreds of additional common open source server applications. The repositories are available in all regions and are accessed via yum update tools, as well as on the [Linux AMI packages site](#). Hosting repositories in each region allows updates to be deployed quickly and without any data transfer charges. The packages can be installed by issuing yum commands, such as the following example:

```
# sudo yum install httpd
```

Access to the Extra Packages for Enterprise Linux (EPEL) repository is configured, but it is not enabled by default. EPEL provides third-party packages in addition to those that are in the Amazon Linux AMI repositories. The third-party packages are not supported by Amazon Web Services.

Important

EC2 instances launched into an Amazon Virtual Private Cloud (VPC) need to have an Internet Gateway attached to the VPC to contact the yum repository. For more information on Amazon VPC, see [Internet Gateways in the Amazon Virtual Private Cloud User Guide](#).

If you find that the Amazon Linux AMI does not contain an application you need, you can simply install the application directly on your Amazon Linux AMI instance. The Amazon Linux AMI uses RPM and yum for package management, and that is likely the simplest way to install new applications. You should always check to see if an application is available in our central Amazon Linux AMI repository first, because many applications are available there. These applications can easily be added to your AMI instance.

To upload your applications onto an Amazon Linux running instance, use `scp` or `sftp` and then configure the application by logging on to your instance. Your applications can also be uploaded during the instance launch by using the `PACKAGE_SETUP` action from built-in the cloud-init package. For more information, see [cloud-init \(p. 73\)](#).

Accessing Source Packages for Reference

You can view the source of packages you have installed inside Amazon EC2 for reference purposes by using tools provided in the Amazon Linux AMI. Source packages are available for all of the packages included in the Amazon Linux AMI and the online package repository. Simply determine the package name for the source package you want to install and use the `get_reference_source` command to view source within your running instance. For example:

```
# get_reference_source -p httpd
```

Following is a sample response:

```
# get_reference_source -p httpd
Requested package: httpd
Found package from local RPM database: httpd-2.2.22-1.24.amzn1.x86_64
```

```
Corresponding source RPM to found package: httpd-2.2.22-1.24.amzn1.src.rpm
```

```
Are these parameters correct? Please type 'yes' to continue: yes
Source RPM downloaded to: /usr/src/srpm/debug/httpd-2.2.22-1.24.amzn1.src.rpm
```

The source RPM will be placed in the /usr/src/srpm/debug directory of your running Amazon EC2 instance. From there it can be unpacked, and, for reference, you can view the source tree using standard RPM tools. After you finish debugging, the package will be available for use in Amazon EC2.

Important

EC2 instances launched into an Amazon Virtual Private Cloud (Amazon VPC) need to have an Internet Gateway attached to the VPC to contact the yum repository. For information on Amazon VPC, see [Internet Gateways](#) in the *Amazon Virtual Private Cloud User Guide*.

Developing Applications

A full set of Linux development tools is provided in the yum repository for the Amazon Linux AMI. To develop applications on the Amazon Linux AMI, simply select the development tools you need with yum. Alternatively, many applications developed on CentOS and other similar distributions should run on the Amazon Linux AMI.

Instance Store Access

The instance store drive ephemeral0 is mounted in /media/ephemeral0 only on Amazon instance store-backed AMIs. This is different than many other images that mount the instance store drive under /mnt.

Product Life Cycle

The Amazon Linux AMI is updated regularly with security and feature enhancements. If you do not need to preserve data or customizations on your running Amazon Linux AMI instances, you can simply relaunch new instances with the latest updated Amazon Linux AMI. If you do need to preserve data or customizations on your running Amazon Linux AMI instances, you can maintain those instances through the Amazon Linux AMI yum repositories. The yum repositories contain all the updated packages. You can chose to apply these updates to your running instances.

Older versions of the AMI and update packages will continue to be available for launch in Amazon EC2, even as new Amazon Linux AMI versions are released. However, in some cases, if you're seeking support for an older version of the Amazon Linux AMI through Amazon Premium Support, we might ask you to move to newer versions as part of the support process.

Important

EC2 instances launched into an Amazon Virtual Private Cloud (Amazon VPC) need to have an Internet Gateway attached to the VPC to contact the yum repository. For information on adding an Internet Gateway to your Amazon VPC, see [Internet Gateways](#) in the *Amazon Virtual Private Cloud User Guide*.

Security Updates

Security updates are provided via the Amazon Linux AMI yum repositories as well as via updated Amazon Linux AMIs. Security alerts will be published in the [Amazon Linux AMI Security Center](#). For more information on AWS security policies or to report a security problem, visit the [AWS Security Center](#).

Amazon Linux AMIs are configured to download and install security updates at launch time. This is controlled via a cloud-init setting called `repo_upgrade`. The following snippet of cloud-init configuration shows how you can change the settings in the user data text you pass to your instance initialization:

```
#cloud-config
repo_upgrade:security
```

The possible values for the `repo_upgrade` setting are as follows:

- `security`
 - Apply outstanding updates that Amazon marks as security updates.
- `bugfix`
 - Apply updates that Amazon marks as bug fixes. Bug fixes are a larger set of updates, which include security updates and fixes for various other minor bugs.
- `all`
 - Apply all applicable available updates, regardless of their classification.
- `none`
 - Do not apply any updates to the instance on startup.

The default setting for `repo_upgrade` is `security`. That is, if you don't specify a different value in your user data, by default the Amazon Linux AMI will perform the security upgrades at launch for any packages installed at that time. Amazon Linux AMI will also notify you of any updates to the installed packages by listing the number of available updates upon login using the `motd`. To install these updates, you will need to run `sudo yum upgrade` on the instance.

Important

EC2 instances launched into an Amazon Virtual Private Cloud (Amazon VPC) need to have an Internet Gateway attached to the VPC to contact the yum repository. For information on adding an Internet Gateway to Amazon VPC, see [Internet Gateways in the Amazon Virtual Private Cloud User Guide](#).

Support

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

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

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

Enabling Your Own Linux Kernels

Topics

- [PV-GRUB: A New Amazon Kernel Image \(p. 78\)](#)
- [Technical Notes for Advanced Users \(p. 83\)](#)
- [Kernel FAQ \(p. 84\)](#)

Amazon EC2 allows you to load a paravirtual Linux kernel within an Amazon Machine Image (AMI) or Amazon EBS volume. You have the option to create images that contain a kernel and initrd (initial RAM disk), and behave in a manner that is closer to traditional virtual or physical Linux installations. By enabling you to boot from the kernel within volumes, this feature allows you to seamlessly upgrade the kernel on Amazon EC2 instances. We expect that AMI providers will update their AMIs to use this new feature, and

most Amazon EC2 users will be able to begin managing their own kernels when these updated AMIs become available. Your AMI provider can tell you when it plans to support this feature. However, if you want to begin managing your own kernel now, the following section shows how. This process assumes general knowledge of Amazon EC2 AMI bundling and registration, as well as knowledge of how to install kernel packages and configure GRUB on your Linux systems.

PV-GRUB: A New Amazon Kernel Image

To enable user-provided kernels, Amazon has published Amazon Kernel Images (AKIs) that use a system called PV-GRUB. PV-GRUB is a paravirtual “mini-OS” that runs a version of GNU GRUB, the standard Linux boot loader. PV-GRUB selects the kernel to boot by reading /boot/grub/menu.lst from your image. It will load the kernel specified by your image and then shut down the “mini-OS,” so that it no longer consumes any resources. One of the advantages of this solution is that PV-GRUB understands standard grub.conf or menu.lst commands, which allows it to work with most existing Linux distributions.

The following task list describes what you need to do to enable an AMI to use PV-GRUB AKI to run a user-provided kernel.

Enabling an AMI to Run A User-Provided Kernel

1. Install an Amazon EC2-compatible kernel.
2. Generate an

initrd

3. Populate

boot/grub/menu.lst

referencing your kernel.

4. Select an appropriate AKI ID from the Amazon Kernel Image IDs section that follows.
5. Bundle the AMI and set the default to your chosen AKI.
6. Upload and register your new AMI.
7. For existing AMIs, you can simply specify the appropriate AKI ID when you call RunInstances or when you use the AWS Management Console.

Note: To update an existing AMI to use a user-provided kernel, re-launch the AMI and follow steps 1 through 6 specified in the preceding task list.

For procedures associated with the preceding task list, see [Using the User-Provided Kernel \(p. 82\)](#).

Configuring the GRUB

In order for PV-GRUB to boot, a GRUB menu.lst file must exist in the image. For most distributions, you have two options for the GRUB configuration:

- **Option 1:** Install GRUB and allow the default kernel installation scripts to handle the installing and updating the GRUB configuration. The steps necessary to install GRUB will vary depending on your Linux distribution, but typically GRUB will be available as a package you can install online.
- **Option 2:** Populate a general /boot/grub/menu.lst. An example of a menu.lst configuration file for booting an AMI with a PV-GRUB AKI follows.

Important

You must modify your own menu.lst for your specific environment.

```
default 0
timeout 3
fallback 1

title Vanilla EC2 Kernel 2.6.32.10
root (hd0)
kernel /boot/vmlinuz-2.6.32.10-ACME_SYS_EC2 root=/dev/sda1
initrd /boot/initrd-2.6.32.10-ACME_SYS_EC2

title Ubuntu EC2 2.6.32.302-EC
root (hd0)
kernel /boot/ubuntu-ec2 root=/dev/sda1
initrd /boot/initrd-ec2
```

We recommend that you use option two to control the kernel booting for two reasons. First, Amazon EC2 users don't have interactive control over the boot process because there is no keyboard access. GRUB will proceed without user interaction. Second, and most important for Amazon EC2 instances, you want to protect against distributions that auto-update the default kernel and break your image. By not relying on the auto-update mechanism and explicitly choosing which kernel you run, you reduce the risk of an incompatible kernel becoming the default kernel.

A fallback kernel does not have to be specified in your menu.lst, but we recommend that you have a fallback when you test new kernels. GRUB can fall back to another kernel in the event that the new kernel fails. Having a fallback kernel allows the instance to boot even if the new kernel is not found.

Amazon EBS Volumes on a PV-GRUB-Enabled AMI

There are two special things you should consider when you use a PV-GRUB-enabled image to mount EBS volumes. First, for Amazon EBS volumes the first partition must be a boot partition. Second, if you plan to use a logical volume manager (LVM) with Amazon EBS volumes, you need a separate boot partition outside of the LVM. Then you can create logical volumes with the LVM. PV-GRUB expects to find the menu.lst in /boot/grub. As a result, if the boot partition is mounted in at /boot, menu.lst will be found in /boot/boot/grub.

Amazon Kernel Image IDs

Several PV-GRUB AKIs are available depending on the type and location of your instance. There are AKIs for 32-bit and 64-bit architecture types, with each having one AKI for partitioned images and another AKI for partitionless images. You must choose an AKI with "hd0" in the name if you want a raw or unpartitioned disk image (most images). Choose an AKI with "hd00" in the name if you want an image that has a partition table.

Most vendors, such as Fedora, Red Hat, Ubuntu, and Novell, use unpartitioned disk images. This means that they use the hd0 variants of PV-GRUB; almost without exception most users will want to use the hd0 variants.

Note

- You cannot use the 64-bit version of PV-GRUB to start a 32-bit kernel or vice versa.
- You must not specify an Amazon ramdisk image (ARI) when using a PV-GRUB AKI.

The following AKI IDs should be used by users who are either registering new AMIs or who want to launch existing AMIs using PV-GRUB. Each AKI type is available in all Amazon EC2 regions:

- **us-east-1**

aki-88aa75e1 pv-grub-hd0_1.03-x86_64.gz

aki-b6aa75df pv-grub-hd0_1.03-i386.gz

aki-b4aa75dd pv-grub-hd00_1.03-x86_64.gz

aki-b2aa75db pv-grub-hd00_1.03-i386.gz

- **us-west-1**

aki-f77e26b2 pv-grub-hd0_1.03-x86_64.gz

aki-f57e26b0 pv-grub-hd0_1.03-i386.gz

aki-eb7e26ae pv-grub-hd00_1.03-x86_64.gz

aki-e97e26ac pv-grub-hd00_1.03-i386.gz

- **us-west-2**

aki-fc37bacc pv-grub-hd0_1.03-x86_64.gz

aki-fa37baca pv-grub-hd0_1.03-i386.gz

aki-f837bac8 pv-grub-hd00_1.03-x86_64.gz

aki-f637bac6 pv-grub-hd00_1.03-i386.gz

- **eu-west-1**

aki-71665e05 pv-grub-hd0_1.03-x86_64.gz

aki-75665e01 pv-grub-hd0_1.03-i386.gz

aki-8b655dff pv-grub-hd00_1.03-x86_64.gz

aki-89655dfd pv-grub-hd00_1.03-i386.gz

- **ap-southeast-1**

aki-fe1354ac pv-grub-hd0_1.03-x86_64.gz

aki-f81354aa pv-grub-hd0_1.03-i386.gz

aki-fa1354a8 pv-grub-hd00_1.03-x86_64.gz

aki-f41354a6 pv-grub-hd00_1.03-i386.gz

- **ap-southeast-2**

aki-3f990e05 pv-grub-hd00_1.03-i386.gz

aki-3d990e07 pv-grub-hd00_1.03-x86_64.gz

aki-33990e09 pv-grub-hd0_1.03-i386.gz

aki-31990e0b pv-grub-hd0_1.03-x86_64.gz

- **ap-northeast-1**

aki-44992845 pv-grub-hd0_1.03-x86_64.gz

aki-42992843 pv-grub-hd0_1.03-i386.gz

aki-40992841 pv-grub-hd00_1.03-x86_64.gz

aki-3e99283f pv-grub-hd00_1.03-i386.gz

- **sa-east-1**

```
aki-c48f51d9 pv-grub-hd0_1.03-x86_64.gz
aki-ca8f51d7 pv-grub-hd0_1.03-i386.gz
aki-c88f51d5 pv-grub-hd00_1.03-x86_64.gz
aki-ce8f51d3 pv-grub-hd00_1.03-i386.gz
• us-gov-west-1
aki-79a4c05a pv-grub-hd0_1.03-x86_64.gz
aki-7ba4c058 pv-grub-hd0_1.03-i386.gz
aki-75a4c056 pv-grub-hd00_1.03-x86_64.gz
aki-77a4c054 pv-grub-hd00_1.03-i386.gz
```

Distribution Kernels Compatible with EC2

There are a number of Linux distributions that have compatible Amazon EC2 kernels. The following is a brief, non-comprehensive list of kernels that we have worked with the maintainers to test:

- Fedora 8-9 Xen kernels
- Fedora 13 (2.6.33.6-147 and higher)
- Fedora 14 and later
- SUSE Linux Enterprise Server
- openSUSE 10.x, 11.0, 11.1 Xen
- openSUSE 11.x EC2 Variant
- Oracle Enterprise Linux
- RedHat Enterprise Linux 5.x kernels
- RedHat Enterprise Linux 6.x kernels
- Ubuntu EC2 Variant kernels
- Ubuntu 11.04 and later
- CentOS 5.x kernels
- CentOS 6.x kernels

It is possible that your specific Linux kernel will not boot using the new PV-GRUB method. If that occurs, select a different kernel or use a non-PV-GRUB AKI to boot your instance.

PV-GRUB Supported File Systems

Since PV-GRUB is a paravirtual version of GRUB 0.97, it has all the limitations of GRUB. Most importantly, this means that it will not work properly for certain disk layouts or file system types. The following are the /boot file systems from which PV-GRUB can boot:

- EXT2/3/4
- XFS
- ReiserFS
- BTRFS (beta)

Note

These are the /boot file systems that we have tested and verified. Others could boot from PV-GRUB, but haven't been tested.

Using the User-Provided Kernel

The following procedure gives an example of how to enable a openSUSE AMI to use the PV-GRUB AKI to run a user-provided kernel by rebundling from a running instance.

Important

The specific details of configuring your AMI to use PV-GRUB will vary depending on your exact Linux environment. The following example is for openSUSE 11.2.

The following procedure uses an instance store-backed AMI, and uploads a bundle from your running instance to Amazon S3. For more information about instance store-backed AMIs, see [Creating Instance Store-Backed Linux/UNIX AMIs \(p. 34\)](#).

If your source AMI is EBS-backed, use `ec2-create-image` to take a snapshot of your AMI and create an image from it. For more information, go to [ec2-create-image](#) in the EC2 CLI Tools Reference and [Creating Amazon EBS-Backed Linux AMIs \(p. 31\)](#).

To use the PV-GRUB AKI with an openSUSE AMI

1. Install an Amazon EC2-compatible kernel from the command line on your running Linux instance.

```
# rpm -ivh /tmp/kernel-ec2-2.6.35-rc4.8.1.x86_64.rpm
warning: /tmp/kernel-ec2-2.6.35-rc4.8.1.x86_64.rpm: Header V3 DSA signature:
NOKEY, key ID a29f6635
Preparing... #####################################################
[100%] #####
1:kernel-ec2 #####
[100%]

Kernel image:      /boot/vmlinux-2.6.35-rc4-8-ec2
Initrd image:     /boot/initrd-2.6.35-rc4-8-ec2
Root device:       /dev/sda1 (mounted on / as ext3)

Features:          block
14807 blocks
```

2. Generate an initrd on your running Linux instance.

```
# mkinitrd
Kernel image:      /boot/vmlinux-2.6.35-rc4-8-ec2
Initrd image:     /boot/initrd-2.6.35-rc4-8-ec2
Root device:       /dev/sda1 (mounted on / as ext3)
Features:          block
14806 blocks
```

3. Populate `/boot/grub/menu.lst` referencing your kernel on your running Linux instance.

Important

Your must modify your own `menu.lst` for your specific environment.

```
default 0
timeout 3
```

```
title EC2
root (hd0)
kernel /boot/vmlinuz-ec2 root=/dev/sda1
initrd /boot/initrd-ec2
```

4. Select an appropriate AKI ID from the Amazon Kernel Image IDs section that follows. For this host, we've chosen aki-88aa75e1 because we are bundling an AMI. Do not specify an Amazon ramdisk image (ARI) when using a PV-GRUB AKI.
5. Bundle the AMI and set the default to your chosen AKI from your running Linux instance. For more information, go to [ec2-bundle-vol](#) in the EC2 AMI Tools Reference.

```
# ec2-bundle-vol -r x86_64 -d /mnt -p openSUSE-11.2-PV-GRUB -u [AWS-ID] -k
/mnt/pkey.pem -c /mnt/cert.pem -s 10240 -e /mnt/root/.ssh --kernel aki-
88aa75e1
```

Note

- X.509 (SSL) certificates are required for bundling operations. To learn more about X.509 (SSL) certificates, such as cert.pem and pkey.pem, go to [X.509 Certificates](#) in *About AWS Security Credentials*.
 - To obtain X.509 certificates, log into the [AWS Management Console](#) (<http://aws.amazon.com/console>) and select [Access Credentials](#), then select the **X.509 Certificates** tab.
6. Register the AMI with the AKI (aki-88aa75e1) from your desktop using ec2-register. For more information, go to [ec2-register](#) in the EC2 CLI Tools Reference.

```
$ ec2-register --name openSUSE-11.2-PVGRUB MyReallyCoolBucketLoca
tion/openSUSE-11.2-PVGRUB.manifest.xml
```

Amazon Support for PV-GRUB

Amazon supports the use of PV-GRUB to load a kernel of your choice for your AMI. However, we cannot provide support for your kernel itself or failures caused by the use of a kernel that does not meet the requirements of PV-GRUB. Due to the wide and varied kernel landscape, it is impossible for Amazon to provide support for all kernel varieties.

Note

The ap-southeast-2 region only provides support for PV-GRUB to load a kernel of your choice from the AMI.

Technical Notes for Advanced Users

The following information is provided to assist those who are familiar with compiling kernels. Many Linux distributions provide documentation on how to compile a kernel in kernel source packages. Amazon is unable to offer support for compiling your own kernel.

It is recommended that you use a PVOPS kernel, release 2.6.39 or later. Release 2.6.39 or later, with PVOPS and Xen options enabled, will work out of the box.

If you are working with 2.6.32 through 2.6.38, boot with the XSAVE instruction in the guest as shown below.

openSUSE

The openSUSE distribution kernel provides native source code that enables compiling an Amazon EC2-capable kernel. For openSUSE, the configuration options are part of the mainline 11.2 and higher. Please use the following kernel configuration options:

```
CONFIG_XEN_COMPAT_00002_AND_LATER=y  
CONFIG_XEN_COMPAT=0x030002  
CONFIG_HOTPLUG_CPU=y
```

Patching vanilla or other distribution kernels (PVOPS)

For PVOPS kernels, you can elect to disable the XSAVE instruction in the guest. The following patch works against 2.6.32 through 2.6.38 kernels.

```
---  
 arch/x86/xen/enlighten.c |    1 +  
 1 files changed, 1 insertions(+), 0 deletions(-)  
  
diff --git a/arch/x86/xen/enlighten.c b/arch/x86/xen/enlighten.c  
index 52f8e19..6db3d67 100644  
--- a/arch/x86/xen/enlighten.c  
+++ b/arch/x86/xen/enlighten.c  
@@ -802,6 +802,7 @@ static void xen_write_cr4(unsigned long cr4)  
 {  
     cr4 &= ~X86_CR4_PGE;  
     cr4 &= ~X86_CR4_PSE;  
+    cr4 &= ~X86_CR4_OSXSAVE;  
  
     native_write_cr4(cr4);  
 }  
--  
1.6.6.1
```

Kernel FAQ

Q: 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.

Q: How do I find user selectable kernels?

Use **ec2-describe-images -o amazon --filter "image-type=kernel"**. This lists all public kernels that are currently available.

Q: What type of dependencies do kernels have?

Kernels are most likely to require a RAM disk that contains required drivers (for example, Xen drivers, video drivers, and so on). If you launch a kernel without a required RAM disk, it will not work properly.

Q: 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 foundation 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 [Paid AMIs \(p. 26\)](#).

Q: Can I use my own kernel?

Yes. For more information, see [Enabling Your Own Linux Kernels \(p. 77\)](#).

Q: How do I know which kernels are compatible with PV-GRUB?

For a list of compatible PV-GRUB kernels, see [Distribution Kernels Compatible with EC2 \(p. 81\)](#). Some Linux distributions provide kernels that are not compatible with Amazon EC2. We are working with vendors to ensure that the most popular AMIs provide kernels that work with Amazon EC2, and we have tested a number of these AMIs and found them to be compatible with PV-GRUB. Unfortunately, it is not possible to support every kernel that is or can be compiled. To avoid the situation in which a kernel does not work consistently or at all, we recommend that you use a known good kernel, select a non-PV-GRUB AKI, or seek support from your AMI vendor.

Q: In what runlevel do instances start?

If you use one of the Xen provided kernels to boot your EC2 instance it will default to run level 4. However if you use PV-GRUB to boot your own kernel inside of the instance, the instance will then default to the internally configured run level.

Q: Which PV-GRUB kernel should I use?

It is recommended that you always use the latest version of the PV-GRUB AKI, as not all versions of the PV-GRUB AKI will be compatible with all instance types. Use the following command to get a list of the PV-GRUB AKIs:

```
$ ec2-describe-images -o amazon --filter "manifest-location=*pv-grub-hd0*"
```

Amazon EC2 Instances

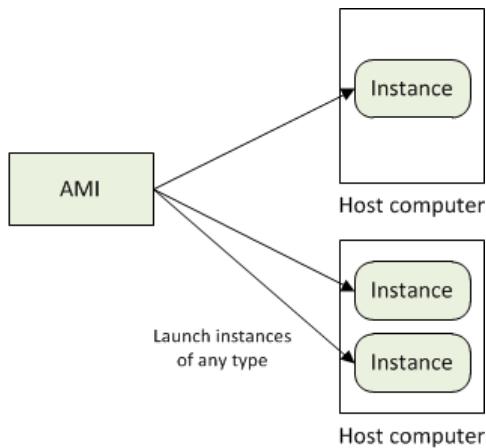
Topics

- [Instance Basics \(p. 87\)](#)
- [Spot Instances \(p. 132\)](#)
- [Reserved Instances \(p. 212\)](#)
- [Launching EC2 Instances \(p. 274\)](#)
- [Connecting to Amazon EC2 Instances \(p. 293\)](#)
- [Importing and Exporting Instances \(p. 308\)](#)
- [Managing Instances \(p. 340\)](#)
- [Stopping Instances \(p. 375\)](#)
- [Troubleshooting Instances \(p. 389\)](#)

This section provides information about instances and their basic characteristics. It also describes how to launch an instance and connect to it.

Amazon EC2 provides templates known as Amazon Machine Images (AMIs) that contains a software configuration (for example, an operating system, an application server, and applications.) You use these templates to launch an *instance*, which is a copy of the AMI running as a virtual server in the cloud.

You can launch different types of instances from a single AMI. An *instance type* essentially determines the hardware of the host computer used for your instance. Each instance type offers different compute and memory capabilities. Select an instance type based on the amount of memory and computing power that you need for the application or software that you plan to run on the instance. For more information, see [Available Instance Types \(p. 88\)](#). You can launch multiple instances from an AMI, as shown in the following figure.



Your instance keeps running until you stop or terminate it, or until it fails. If an instance fails, you can launch a new one from the AMI.

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

Related Topics

- [Amazon Machine Images \(AMI\) \(p. 18\)](#)
- [Amazon EC2 Instance Store \(p. 513\)](#)

Instance Basics

Topics

- [Instance Families and Types \(p. 87\)](#)
- [Regions and Availability Zones \(p. 116\)](#)
- [Supported Platforms \(p. 120\)](#)
- [Amazon EC2 Root Device Volume \(p. 123\)](#)
- [Instance Best Practices \(p. 129\)](#)
- [Stopping, Starting, and Terminating Instances \(p. 130\)](#)

This section provides information about key concepts to help you understand Amazon EC2 instances.

Instance Families and Types

Topics

- [Available Instance Types \(p. 88\)](#)
- [EBS-Optimized Instances \(p. 91\)](#)
- [Private IP Addresses Per ENI Per Instance Type \(p. 91\)](#)
- [Windows Instance Types \(p. 92\)](#)
- [Compute Resources Measurement \(p. 93\)](#)
- [I/O Resources \(p. 93\)](#)
- [Instance Tags \(p. 94\)](#)
- [Micro Instances \(p. 94\)](#)

- [High I/O Instances \(p. 101\)](#)
- [High Storage Instances \(p. 102\)](#)
- [Cluster Instances \(p. 104\)](#)

Amazon Elastic Compute Cloud (Amazon EC2) instances are grouped into the general families described in the following table.

Family	Description
Cluster Compute	Have a very large amount of CPU coupled with increased networking performance. They're well-suited for High Performance Compute (HPC) applications and other demanding network-bound applications. For more information, see Overview (p. 104) .
Cluster GPU	Provide general-purpose graphics processing units (GPUs), with proportionally high CPU and increased network performance for applications that benefit from highly parallelized processing. They're well-suited for HPC applications as well as rendering and media processing applications. For more information, see Overview (p. 104) .
High CPU	Have proportionally more CPU resources than memory (RAM). They're well-suited for compute-intensive applications.
High I/O	Provide tens of thousands of low-latency, random I/O operations per second (IOPS) to an application. They're well-suited for NoSQL databases, clustered databases, and OLTP (online transaction processing) systems. For more information, see High I/O Instances (p. 101) .
High Memory	Have proportionally more memory resources. They're well suited for high-throughput applications, such as database and memory caching applications.
High-Memory Cluster	Have large amounts of memory coupled with high CPU and network performance. These instances are well suited for in-memory analytics, graph analysis, and scientific computing applications.
High Storage	Provide very high storage density and high sequential read and write performance per instance. They are well-suited for data warehousing, Hadoop/MapReduce, and parallel file systems. For more information, see High Storage Instances (p. 102) .
Micro	Provide a small amount of consistent CPU resources and enable you to burst CPU capacity when additional cycles are available. They're well-suited for lower throughput applications and websites that consume significant compute cycles periodically. For more information, see Micro Instances (p. 94) .
Standard	Have memory-to-CPU ratios suitable for most general-purpose applications.

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

When you launch an instance, you specify the instance type. If you don't specify an instance type when you launch an instance, the default is an M1 Small (`m1.small`) instance.

Name	Memory	Compute Units	Virtual Cores	Instance Store Volumes*	Architecture	Network Performance	Available for Spot Instance	API Name
Cluster Compute								
Cluster Compute Eight Extra Large	60.5 GiB	88	16 (2 x Intel Xeon E5-2670, eight-core with hyperthread)	3360 GB (4 x 840 GB)	64-bit	Very high (10 Gbps Ethernet)	Yes	cc2.8xlarge
Cluster Compute Quadruple Extra Large	22.5 GiB	33.5	8 (2 x Intel Xeon X5570, quad-core with hyperthread)	1680 GB (2 x 840 GB)	64-bit	Very high (10 Gbps Ethernet)	Yes	cc1.4xlarge
Cluster GPU								
Cluster GPU Quadruple Extra Large**	22.5 GiB (see note after this table)	33.5	8 (2 x Intel Xeon X5570, quad-core with hyperthread), plus 2 NVIDIA Tesla M2050 GPUs	1680 GB (2 x 840 GB)	64-bit	Very high (10 Gbps Ethernet)	Yes	cgl.4xlarge
High CPU								
High-CPU Extra Large	7 GiB	20	8 (with 2.5 ECUs each)	1680 GB (4 x 420 GB)	64-bit	High	Yes	c1.xlarge
High-CPU Medium	1.7 GiB	5	2 (with 2.5 ECUs each)	350 GB (1 x 350 GB)	32-bit and 64-bit	Moderate	Yes	c1.medium
High I/O								
High I/O Quadruple Extra Large***	60.5 GiB	35	8 (with 4.37 ECUs each)	2,048 GB (2 x 1,024 GB) SSD	64-bit	Very high (10 Gbps Ethernet)	No	hi1.4xlarge
High Memory								
High Memory Double Extra Large	34.2 GiB	13	4 (with 3.25 ECUs each)	850 GB (1 x 850 GB)	64-bit	Moderate	Yes	m2.2xlarge

Name	Memory	Compute Units	Virtual Cores	Instance Store Volumes*	Architecture	Network Performance	Available for Spot Instance	API Name
High Memory Extra Large	17.1 GiB	6.5	2 (with 3.25 ECUs each)	420 GB (1 x 420 GB)	64-bit	Moderate	Yes	m2.xlarge
High Memory Quadruple Extra Large	68.4 GiB	26	8 (with 3.25 ECUs each)	1680 GB (2 x 840 GB)	64-bit	High	Yes	m2.4xlarge
High-Memory Cluster								
High Memory Cluster Eight Extra Large	244 GiB	88	16 (2 x Intel Xeon E5-2670, eight-core)	240 GB (2 x 120 GB) SSD	64-bit	Very high (10 Gbps Ethernet)	Yes	cr1.8xlarge
High Storage								
High Storage Eight Extra Large	117 GiB	35	16 (8 cores + 8 hyperthreads)	49,152 GB (24 x 2,048 GB) hard disk drives	64-bit	Very high (10 Gbps Ethernet)	No	hs1.8xlarge
Micro								
Micro	615 MiB	Up to 2 (for short periodic bursts)	1	EBS storage only	32-bit and 64-bit	Very low	Yes	t1.micro
Standard								
M1 Extra Large	15 GiB	8	4 (with 2 ECUs each)	1680 GB (4 x 420 GB)	64-bit	High	Yes	m1.xlarge
M1 Large	7.5 GiB	4	2 (with 2 ECUs each)	840 GB (2 x 420 GB)	64-bit	Moderate	Yes	m1.large
M1 Medium	3.75 GiB	2	1	410 GB (1 x 410 GB)	32-bit and 64-bit	Moderate	Yes	m1.medium
M1 Small	1.7 GiB	1	1	160 GB (1 x 160 GB)	32-bit and 64-bit	Low	Yes	m1.small
M3 Double Extra Large	30 GiB	26	8 (with 3.25 ECUs each)	EBS storage only	64-bit	High	Yes	m3.2xlarge

Name	Memory	Compute Units	Virtual Cores	Instance Store Volumes*	Architecture	Network Performance	Available for Spot Instance	API Name
M3 Extra Large	15 GiB	13	4 (with 3.25 ECUs each)	EBS storage only	64-bit	Moderate	Yes	m3.xlarge

*Instance store volumes may not be mounted by default. For more information, see [Instance Store Device Names \(p. 515\)](#).

**The cg1.4xlarge instance type has 1 GiB reserved for GPU operation. The 21.5 GiB doesn't include the on-board memory of the GPUs, which is 3 GiB per GPU for the NVIDIA Tesla M2050. Before you launch your first cg1.4xlarge instance, complete the [Amazon EC2 instance request form](#).

***The hi1.4xlarge instance type is based on solid-state drive (SSD) technology. For more information, see [High I/O Instances \(p. 101\)](#).

EBS-Optimized Instances

An EBS-optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for EBS I/O. This optimization provides the best performance for your EBS volumes by minimizing contention between EBS I/O and other traffic from your Amazon EC2 instance.

EBS optimization enables instances to fully utilize the IOPS provisioned on an EBS volume. EBS-optimized instances deliver dedicated throughput to EBS, with options between 500 Mbps and 1,000 Mbps, depending on the instance type you use. When attached to an EBS-optimized instance, Provisioned IOPS volumes are designed to deliver within 10 percent of their provisioned performance 99.9 percent of the time in a given year. For more information, see [Provisioned IOPS Volumes \(p. 458\)](#).

The following instance types can be launched as EBS-optimized instances:

- High-CPU Extra Large (c1.xlarge)
- M1 Large (m1.large)
- M1 Extra Large (m1.xlarge)
- High-Memory Double Extra Large (m2.2xlarge)
- High-Memory Quadruple Extra Large (m2.4xlarge)
- M3 Extra Large (m3.xlarge)
- M3 Double Extra Large (m3.2xlarge)

When you use an EBS-optimized instance, you pay an additional low, hourly fee for the dedicated capacity.

Private IP Addresses Per ENI Per Instance Type

The following table lists the maximum number of Elastic Network Interfaces (ENI) per EC2 instance type and the maximum number of private IP addresses per ENI. ENIs and multiple private IP addresses are only available in EC2 instances running in Amazon Virtual Private Cloud. For more information, see [Multiple IP Addresses \(p. 433\)](#).

Type	Name	Elastic Network Interfaces (ENI)	Private IP Addresses per ENI
Cluster Compute Eight Extra Large	cc2.8xlarge	8	30
Cluster Compute Quadruple Extra Large	cc1.4xlarge	N/A	N/A
Cluster GPU Quadruple Extra Large	cg1.4xlarge	8	30
High-CPU Extra Large	c1.xlarge	4	15
High-CPU Medium	c1.medium	2	6
High I/O Quadruple Extra Large	hi1.4xlarge	8	30
High-Memory Double Extra Large	m2.2xlarge	4	30
High-Memory Extra Large	m2.xlarge	4	15
High-Memory Quadruple Extra Large	m2.4xlarge	8	30
High-Memory Cluster Eight Extra Large	cr1.8xlarge	8	30
High Storage Eight Extra Large	hs1.8xlarge	8	30
M1 Extra Large	m1.xlarge	4	15
M1 Large	m1.large	3	10
M1 Medium	m1.medium	2	6
M1 Small	m1.small	2	4
M3 Double Extra Large	m3.2xlarge	4	30
M3 Extra Large	m3.xlarge	4	15
Micro	t1.micro	2	2

Windows Instance Types

Amazon EC2 instances can run Microsoft Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, or Windows Server 2012. The Windows AMIs provide you with all standard Windows Server functionality.

Using Amazon EC2 instances running Windows is similar to using instances running Linux/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 (available through the AWS Management Console, the command line tools, or the Amazon EC2 API).
- **Bundling**—Amazon instance store-backed Windows instances requires different bundling procedures than Amazon instance store-backed Linux/UNIX instances. For more information, see [Creating an Instance Store-Backed Windows AMI](#) in the *Amazon Elastic Compute Cloud Microsoft Windows Guide*.

Amazon EC2 currently provides the following Windows Server AMIs:

- Microsoft Windows Server 2003 (32-bit)
- Microsoft Windows Server 2003 (64-bit)
- Microsoft Windows Server 2008 (32-bit)
- Microsoft Windows Server 2008 (64-bit)
- Microsoft Windows Server 2008 R2 (64-bit)
- Microsoft Windows Server 2012 (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. For more information about how to determine the internal DNS name, see [Amazon EC2 Instance IP Addressing \(p. 430\)](#).
- 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.

For more information about the EC2 configuration service, see [Using Ec2Config](#) in the *Amazon Elastic Compute Cloud Microsoft Windows Guide*.

Compute Resources Measurement

Changing 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. One EC2 Compute Unit provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor. This is also the equivalent to an early-2006 1.7 GHz Xeon processor referenced in our original documentation.

Note

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

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

I/O Resources

Amazon EC2 provides virtualized server instances. Whereas some resources like CPU, memory, and instance storage are dedicated to a particular instance, other resources such as the network and the disk subsystem are shared among 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 (low, moderate, high, etc.). Instance types with high I/O performance have a larger allocation of shared resources. Allocating a larger share of shared resources also reduces the variance of I/O performance. For most applications, moderate I/O performance is more than enough. However, for applications that require greater or more consistent I/O performance, consider instances with higher I/O performance.

Instance Tags

To help categorize and manage your instances, you can assign *tags* of your choice to them. For more information, see [Tagging Your Amazon EC2 Resources \(p. 540\)](#).

Micro Instances

Topics

- [Hardware Specifications \(p. 94\)](#)
- [Optimal Application of Micro Instances \(p. 94\)](#)
- [Available CPU Resources During Spikes \(p. 96\)](#)
- [When the Instance Uses Its Allotted Resources \(p. 96\)](#)
- [Comparison with the m1.small Instance Type \(p. 98\)](#)
- [AMI Optimization for Micro Instances \(p. 100\)](#)
- [Related Topics \(p. 101\)](#)

Micro instances (t1.micro) provide a small amount of consistent CPU resources and allow you to increase CPU capacity in short bursts when additional cycles are available. They are well suited for lower throughput applications and websites that require additional compute cycles periodically.

This section describes how micro instances generally work so you can understand how to apply them. Our intent is not to specify exact behavior, but to give you visibility into the instance's behavior so you can understand its performance (to a greater degree than you typically get with, for example, a multi-tenant shared web hosting system).

The micro instance is available on both 32-bit and 64-bit platforms, but it's available as an Amazon EBS-backed instance only. For more information about basic specifications for the micro instance type, see [Available Instance Types \(p. 88\)](#).

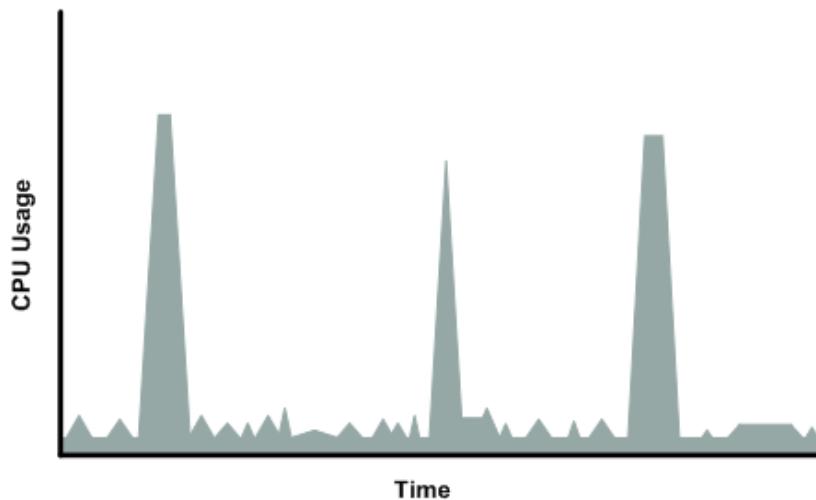
Hardware Specifications

The hardware specifications for t1.micro instances are described in the following table.

Item	Description
Processor	Up to 2 EC2 compute units (for short periodic bursts).
Virtual cores	1
Memory	615 MiB
Platform	32-bit and 64-bit

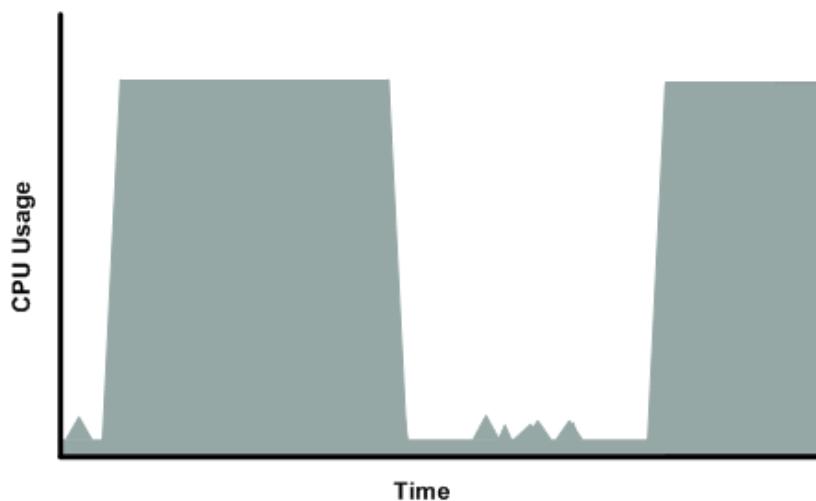
Optimal Application of Micro Instances

Micro instances provide spiky CPU resources for workloads that have a CPU usage profile similar to what is shown in the following figure.

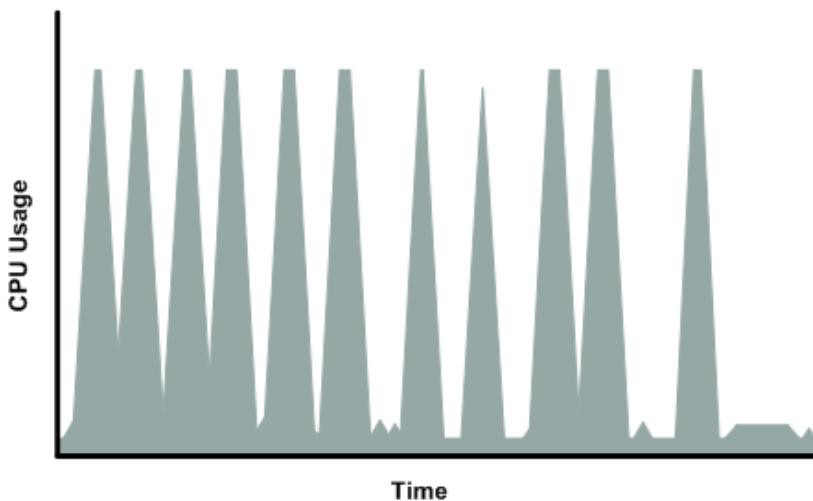


The instance is designed to operate with its CPU usage at essentially only two levels: the normal low background level, and then at brief spiked levels much higher than the background level. We allow the instance to operate at up to 2 EC2 compute units (ECUs) (for more information, see [Compute Resources Measurement \(p. 93\)](#)). The ratio between the maximum level and the background level is designed to be large. Micro instances are designed to support tens of requests per minute on your application. However, actual performance can vary significantly depending on the amount of CPU resources required for each request on your application.

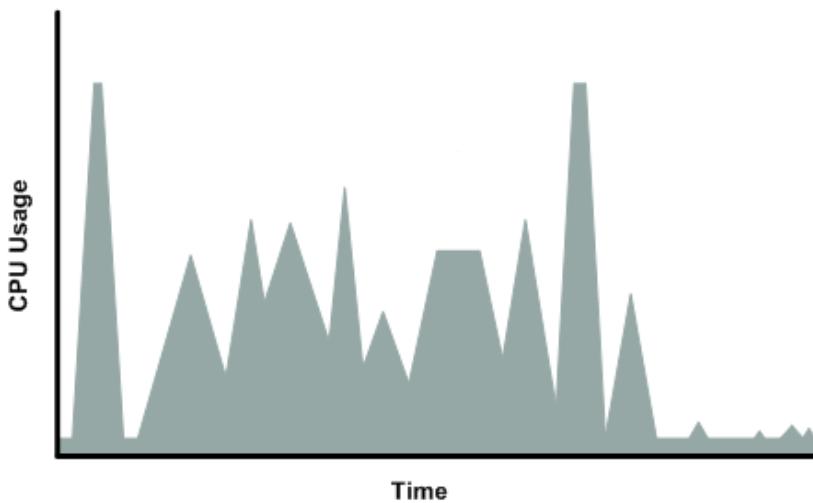
Your application might have a different CPU usage profile than that described in the preceding section. The next figure shows the profile for an application that isn't appropriate for a micro instance. The application requires continuous data-crunching CPU resources for each request, resulting in plateaus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



The next figure shows another profile that isn't appropriate for a micro instance. Here the spikes in CPU use are brief, but they occur too frequently to be serviced by a micro instance.



The next figure shows another profile that isn't appropriate for a micro instance. Here the spikes aren't too frequent, but the background level between spikes is too high to be serviced by a micro instance.



In each of the preceding cases of workloads not appropriate for a micro instance, we recommend that you consider using a different instance type. For more information about instance types, see [Instance Families and Types \(p. 87\)](#).

Available CPU Resources During Spikes

When your instance *bursts* to accommodate a spike in demand for compute resources, it uses unused resources on the host. The amount available depends on how much contention there is when the spike occurs. The instance is never left with zero CPU resources, whether other instances on the host are spiking or not.

When the Instance Uses Its Allotted Resources

We expect your application to consume only a certain amount of CPU resources in a period of time. If the application consumes more than your instance's allotted CPU resources, we temporarily limit the

instance so it operates at a low CPU level. If your instance continues to use all of its allotted resources, its performance will degrade. We will increase the time that we limit its CPU level, thus increasing the time before the instance is allowed to burst again.

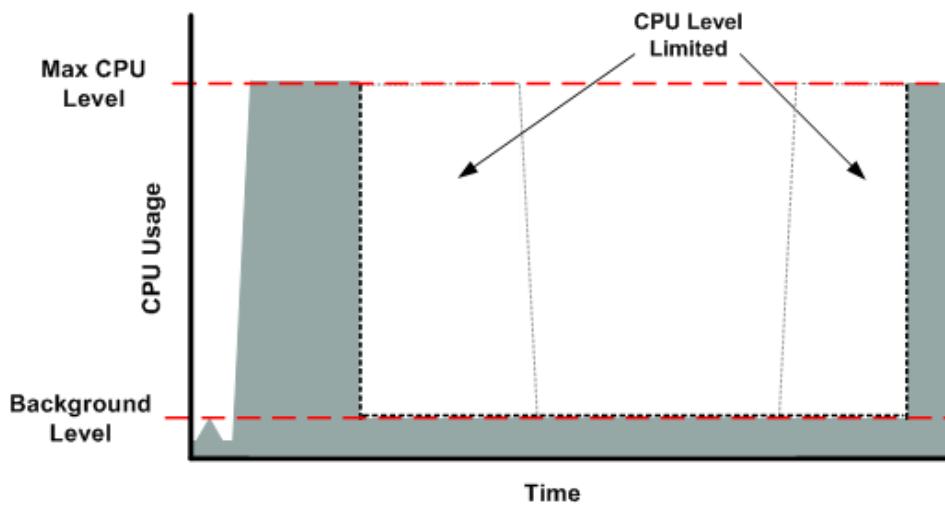
If you enable Amazon CloudWatch monitoring for your micro instance, you can use the "Avg CPU Utilization" graph in the AWS Management Console to determine whether your instance is regularly using all its allotted CPU resources. We recommend that you look at the maximum value reached during each given period. If the maximum value is 100%, we recommend that you use Auto Scaling to scale out (with additional micro instances and a load balancer), or move to a larger instance type. For more information about Auto Scaling, see the [Auto Scaling Developer Guide](#).

Note

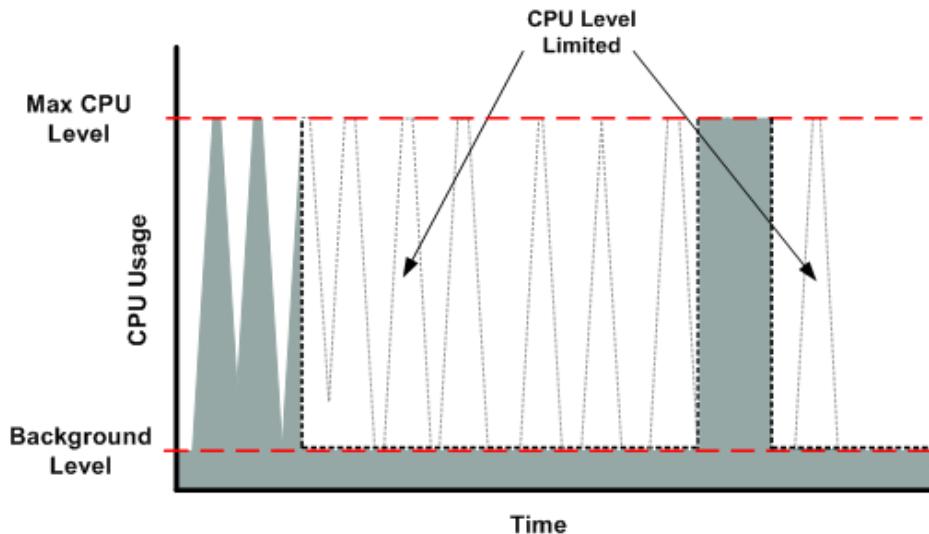
Micro instances are available as either 32-bit or 64-bit. If you need to move to a larger instance, make sure to move to a compatible instance type.

The following figures show the three suboptimal profiles from the preceding section and what it might look like when the instance consumes its allotted resources and we have to limit its CPU level. If the instance consumes its allotted resources, we restrict it to the low background level.

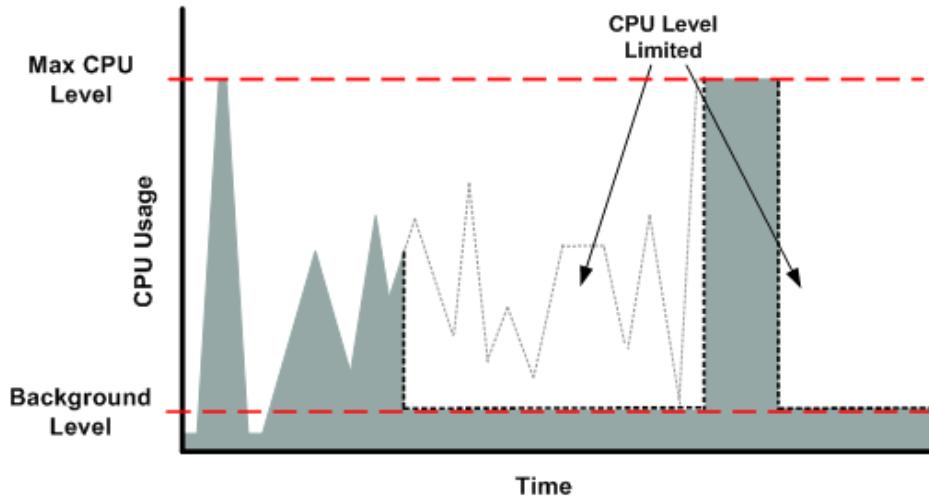
The next figure shows the situation with the long plateaus of data-crunching CPU usage. The CPU hits the maximum allowed level and stays there until the instance's allotted resources are consumed for the period. At that point, we limit the instance to operate at the low background level, and it operates there until we allow it to burst above that level again. The instance again stays there until the allotted resources are consumed and we limit it again (not seen on the graph).



The next figure shows the situation where the requests are too frequent. The instance uses its allotted resources after only a few requests and so we limit it. After we lift the restriction, the instance maxes out its CPU usage trying to keep up with the requests, and we limit it again.

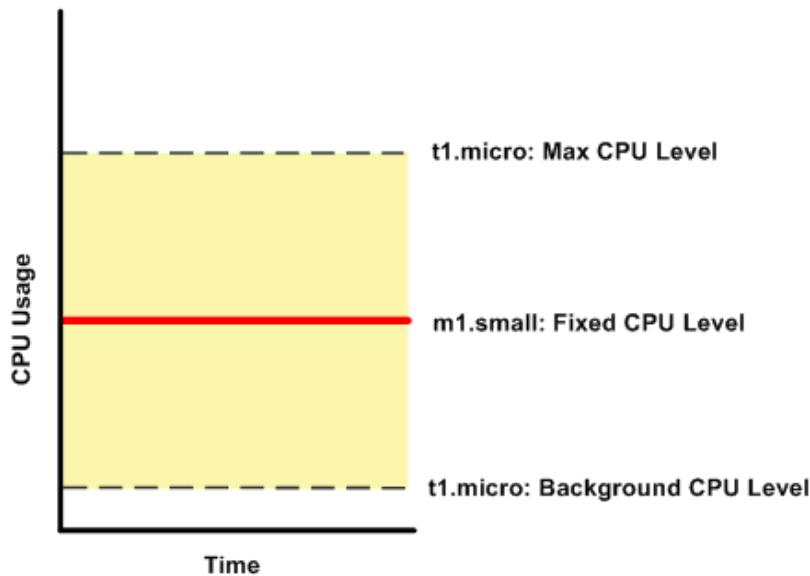


The next figure shows the situation where the background level is too high. Notice that the instance doesn't have to be operating at the maximum CPU level for us to limit it. We limit the instance when it's operating above the normal background level and has consumed its allotted resources for the given period. In this case (as in the preceding one), the instance can't keep up with the work, and we limit it again.



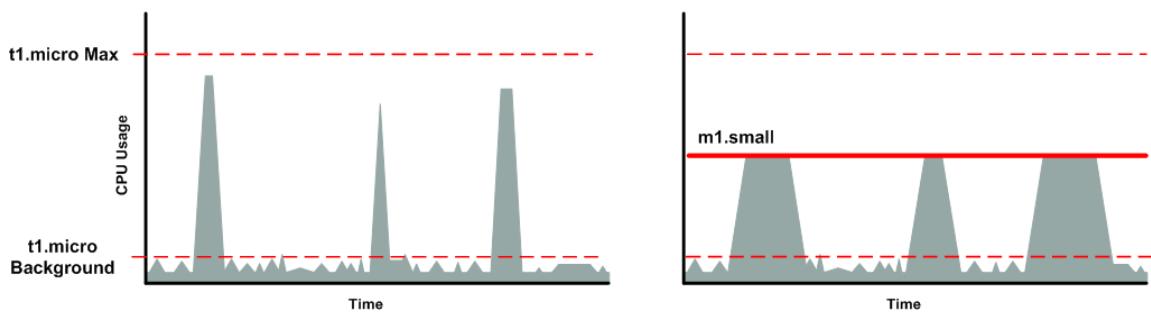
Comparison with the m1.small Instance Type

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

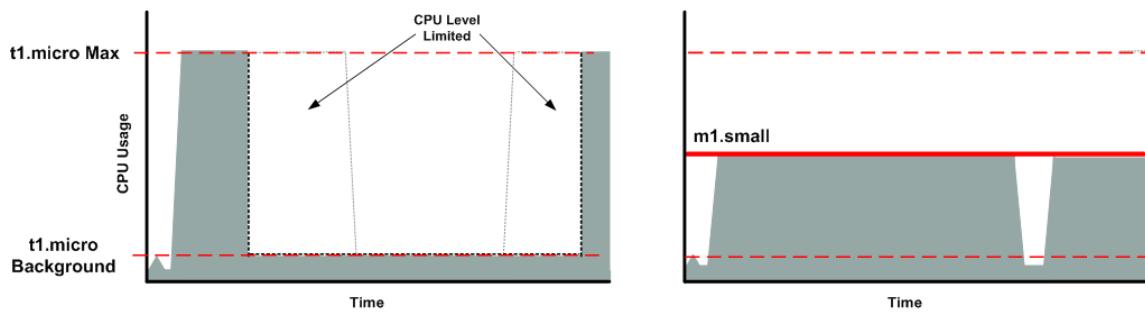


The following figures compare the CPU usage of a micro instance with an m1.small instance for the various scenarios we've discussed in the preceding sections.

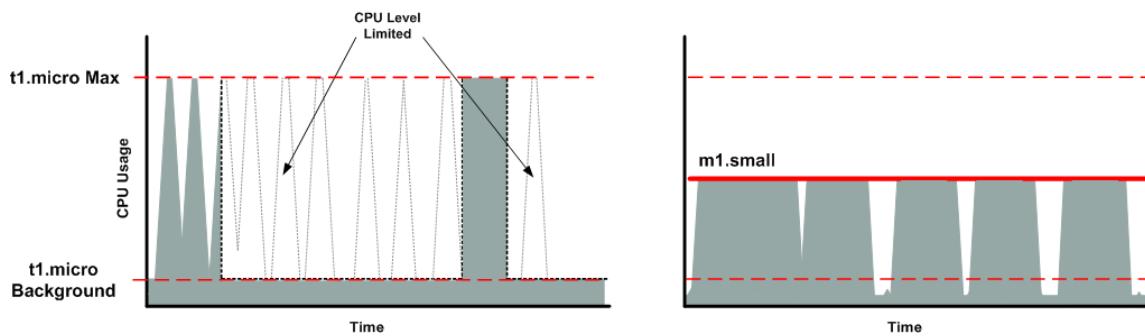
The first figure that follows shows an optimal scenario for a micro instance (the left graph) and how it might look for an m1.small instance (the right graph). In this case, we don't need to limit the micro instance. The processing time on the m1.small instance would be longer for each spike in CPU demand compared to the micro instance.



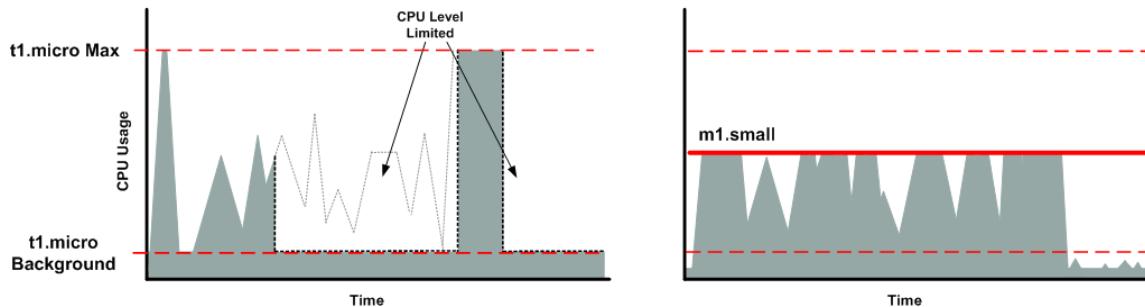
The next figure shows the scenario with the data-crunching requests that used up the allotted resources on the micro instance, and how they might look with the m1.small instance.



The next figure shows the frequent requests that used up the allotted resources on the micro instance, and how they might look on the m1.small instance.



The next figure shows the situation where the background level used up the allotted resources on the micro instance, and how it might look on the m1.small instance.



AMI Optimization for Micro Instances

We recommend that you follow these best practices when optimizing an AMI for the micro instance type:

- Design the AMI to run on 600 MB of RAM
- Limit the number of recurring processes that use CPU time (e.g., cron jobs, daemons)

In Linux, you can optimize performance using swap space and virtual memory (e.g., by setting up swap space in a separate partition from the root file system).

In Windows, when you perform significant AMI or instance configuration changes (e.g., enable server roles or install large applications), you might see limited instance performance, because these changes

can be memory intensive and require long-running CPU resources. We recommend that you first use a larger instance type when performing these changes to the AMI, and then run the AMI on a micro instance for normal operations.

Related Topics

- [Instance Families and Types \(p. 87\)](#)

High I/O Instances

High I/O instances (`hi1.4xlarge`) can deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications. They are well suited for the following scenarios:

- NoSQL databases (for example, Cassandra and MongoDB)
- Clustered databases
- Online transaction processing (OLTP) systems

By default, you can run up to two high I/O instances. If you need more than two high I/O instances, you can request more using the [Amazon EC2 Instance Request Form](#).

You can cluster high I/O instances in a cluster placement group. The placement group cannot contain other cluster instances.

Hardware Specifications

The `hi1.4xlarge` instance type is based on solid-state drive (SSD) technology.

The hardware specifications for `hi1.4xlarge` instances are described in the following table.

Item	Description
Processor	35 EC2 compute units
Virtual cores	16 virtual cores (8 cores + 8 hyperthreads)
Memory	60.5 GiB
Platform	64-bit
Network	10 Gbps Ethernet
Instance store volumes	2 x 1,024 GB SSD

Disk I/O Performance

Using Linux paravirtual (PV) AMIs, high I/O instances can deliver more than 120,000 4 KB random read IOPS and between 10,000 and 85,000 4 KB random write IOPS (depending on active logical block addressing span) to applications across two SSD data volumes. For hardware virtual machines (HVM) and Windows AMIs, performance is approximately 90,000 4 KB random read IOPS and between 9,000 and 75,000 4 KB random write IOPS.

The maximum sequential throughput on all AMI types (Linux PV, Linux HVM, and Windows) per second is approximately 2 GB read and 1.1 GB write.

SSD Storage

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

- The primary data source is an instance store with SSD storage.
- Read performance is consistent and write performance can vary.
- Write amplification can occur.
- The TRIM command is not currently supported.

Instance Store with SSD Storage

High I/O instances use an EBS-backed root device. However, their primary data storage is provided by the SSD volumes in the instance store. Like other instance store volumes, these instance store volumes persist only for the life of the instance. Because the root device of the high I/O instance is EBS-backed, you can still start and stop your instance. When you stop an instance, your application persists, but your production data in the instance store does not persist. For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 513\)](#).

Variable Write Performance

Write performance depends on how your applications utilize logical block addressing (LBA) space. If your applications use the total LBA space, write performance can degrade by about 90 percent. Benchmark your applications and monitor the queue depth (the number of pending I/O requests for a volume) and I/O size.

Write Amplification

Write amplification refers to an undesirable condition associated with flash memory and SSDs, where the actual amount of physical information written is a multiple of the logical amount intended to be written. Because flash memory must be erased before it can be rewritten, the process to perform these operations results in moving (or rewriting) user data and metadata more than once. This multiplying effect increases the number of writes required over the life of the SSD, which shortens the time that it can reliably operate. High I/O instances are designed with a provisioning model intended to minimize write amplification.

Random writes have a much more severe impact on write amplification than serial writes. If you are concerned about write amplification, allocate less than the full tebibyte of storage for your application (also known as over provisioning).

The TRIM Command

The TRIM command enables the operating system to notify an SSD that blocks of previously saved data are considered no longer in use. TRIM limits the impact of write amplification.

TRIM support is not available for high I/O instances at this time. We hope to add TRIM support in the future.

Related Topics

- [Instance Families and Types \(p. 87\)](#)

High Storage Instances

Amazon Elastic Compute Cloud (Amazon EC2) high storage instances (`hs1.8xlarge`) provide very high storage density and high sequential read and write performance per instance. They are well suited for the following scenarios:

- Data warehousing
- Hadoop/MapReduce
- Parallel file systems

By default, you can run up to two high storage instances. If you need more than two high storage instances, you can request more using the [Amazon EC2 Instance Request Form](#).

You can cluster high storage instances in a cluster placement group. The placement group cannot contain other cluster instances.

Hardware Specifications

High storage instances support both EBS-backed and instance store-backed Amazon Machine Images (AMIs). High storage instances support both paravirtual (PV) and hardware virtual machine (HVM) AMIs. High storage instances are capable of delivering 2.4 GB per second of sequential read and 2.6 GB per second of sequential write performance when using a block size of 2 MiB.

For customers using Microsoft Windows Server, high storage instances are only supported with the Microsoft Windows Server AMIs for Cluster Instance Type. Cluster compute, cluster GPU, high I/O, and high storage instances do not currently support EBS optimization, but provide high bandwidth networking and can also be used with Amazon Elastic Block Store (EBS) provisioned I/O operations per second (IOPS) volumes for improved consistency and performance.

For customers using Linux, Linux kernel version 3.3 and higher contain a bug that can cause instability, crashes, and possible data corruption under high I/O load. A [fix for this bug](#) has been submitted to the Linux kernel. We recommend using AMIs based on Linux kernel version 3.3 or lower for hs1.8xlarge instances until the issue has been addressed in the stable kernel releases.

The hardware specifications for the hs1.8xlarge instance type are described in the following table.

Item	Description
Processor	35 EC2 compute units
Virtual cores	16 virtual cores (8 cores + 8 hyperthreads)
Memory	117 GiB
Platform	64-bit
Network	10 Gbps Ethernet
Instance store volumes	24 x 2,048 GB hard disk drives

Disk I/O Performance

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

Storage Information

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

Instance Store with High Storage Instances

High storage instances support both Amazon Simple Storage Service (Amazon S3) and EBS-backed root devices. However, even when using an EBS-backed root device, primary data storage is provided by the hard disk drives in the instance store. Like other instance store volumes, these instance store volumes persist only for the life of the instance. Therefore, when you stop an instance (when using an EBS-backed root volume), your application persists, but your production data in the instance store does not persist. For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 513\)](#).

Disk Initialization

If you plan to run your hs1.8xlarge instance in a steady state for long periods of time, we recommend that you zero the hard disks first for improved performance. This process can take as long as six hours to complete.

Setting the Memory Limit

Many Linux-based Amazon Machine Images (AMIs) come with `CONFIG_XEN_MAX_DOMAIN_MEMORY` set to 70. We recommend that you set this as appropriate for 117 GiB of memory.

Setting the User Limit (ulimit)

Many Linux-based Amazon Machine Images (AMIs) come with a default ulimit of 1024. We recommend that you increase the ulimit to 2048.

Related Topics

- [Instance Families and Types \(p. 87\)](#)

Cluster Instances

In this topic, you'll learn about cluster instances and how to create hardware virtual machine (HVM) AMIs. You'll also learn how to create placement groups and then launch cluster instances into them.

Topics

- [Overview \(p. 104\)](#)
- [How to Launch Cluster Instances in a Placement Group \(p. 106\)](#)
- [How to Delete a Placement Group \(p. 111\)](#)
- [AWS CloudFormation Templates for Cluster Instances \(p. 113\)](#)
- [Using the NVIDIA Driver on a Cluster GPU Instance \(p. 113\)](#)
- [Cluster Instance Hardware Specifications \(p. 114\)](#)

Overview

Amazon Elastic Compute Cloud (Amazon EC2) offers two cluster instance types: cluster compute instances and cluster graphics processing unit (GPU) instances. These instance types provide a very large amount of computational power coupled with increased network performance. They are well suited for High Performance Computing (HPC) applications and other demanding network-bound applications, such as science and engineering applications, financial modeling applications, and business analytics applications.

Cluster GPU instances have the same networking and placement group properties as cluster compute instances, but also include NVIDIA Tesla M2050 GPUs. These GPUs can be used to accelerate many scientific, engineering, and rendering applications by leveraging the Compute Unified Device Architecture (CUDA) or OpenCL parallel computing frameworks.

You can logically group cluster instances into clusters, known as *cluster placement groups*. This enables applications to get the full-bisection bandwidth and low-latency network performance required for tightly coupled, node-to-node communication typical of many HPC applications. A cluster placement group is a logical grouping of cluster instances within a single Availability Zone. A placement group cannot span multiple Availability Zones. You first create a placement group and then launch multiple cluster instances into the placement group. You must give each placement group a name that is unique within your AWS account.

Important

We recommend that you launch the number of instances that you need in the placement group in a single launch request. If you try to add more instances to the placement group later, you increase your chances of getting an "Insufficient Capacity" error. If you do receive an "Insufficient Capacity" error, stop and restart the instances in the placement group, and try to add more instances again.

Tip

You can't merge cluster placement groups. Instead, you must terminate the instances in one group, and then relaunch those instances into the other group.

A cluster instance is an Amazon EC2 instance type. This means that cluster instances interoperate with other Amazon EC2 features and AWS cloud services just as other Amazon EC2 instance types do. The functionality of cluster instances is similar to the functionality of other Amazon EC2 instance types, but with the important differences described in the following table and sections.

Feature	Description
Cluster Placement Groups	Provides low latency and high-bandwidth connectivity between the cluster instances within a single Availability Zone. You just create a placement group and then launch cluster instances into it. Instances launched within a placement group participate in a full-bisection bandwidth cluster appropriate for HPC applications.
HVM-Based Virtualization	Cluster instances run as hardware virtual machine (HVM)-based instances. HVM virtualization uses hardware-assist technology provided by the AWS platform. With HVM virtualization, the guest VM runs as if it were on a native hardware platform, except that it still uses paravirtual (PV) network and storage drivers for improved performance. For more information, see HVM-Based AMIs (p. 106) .
Amazon EBS-Backed, HVM-Based AMIs	You must launch cluster instances from Amazon EBS-backed, HVM-based AMIs. Launching an instance from an Amazon EC2 instance store-backed AMI is not supported. AWS labels all AMIs as either <code>hvm</code> or <code>paravirtual</code> so you can determine the virtualization type. For information about using Amazon EBS-backed AMIs, see Storage for the Root Device (p. 19) .

Current Limitations

Cluster instances currently have the following limitations:

- Cluster instance types aren't available in every region. For more information, see the [Amazon EC2 Release Notes](#).
- Before you launch your first `cg1.4xlarge` instance, complete the [Amazon EC2 instance request form](#).
- Cluster instance types aren't available for use with Amazon DevPay.

Cluster placement groups currently have the following limitations:

- Amazon EC2 Reserved Instances are not guaranteed within a specific requested placement group.

- A cluster placement group can't span multiple Availability Zones.
- A cluster placement group can't contain cluster instances of different instance types.

HVM-Based AMIs

Cluster instances require the use of HVM-based, Amazon EBS-backed AMIs. Paravirtual (PV) AMIs employed by Linux instances and other instance types that use paravirtual machine (PVM)-based virtualization are not supported. Cluster instances do not require the use of Amazon Kernel Images (AKIs) or Amazon RAM disk Images (ARIs) that are part of PVM-based AMIs used with other EC2 instance types.

Reference AMIs

To help you get started with cluster instances, AWS provides a reference Amazon Linux AMI and a reference Amazon Windows AMI. These reference AMIs include the NVIDIA driver and CUDA toolkit, which enable the NVIDIA GPUs. This topic shows you how to customize a cluster instance and create your own AMI from it. To learn more about the Amazon Linux reference AMI, see [Amazon Linux AMI](#). To learn more about the Amazon Windows AMIs, see [Amazon Machine Images \(AMIs\)](#).

How to Launch Cluster Instances in a Placement Group

You can use the following steps to create, customize, and launch cluster instances in a placement group:

1. Launch a cluster instance.
2. Customize your cluster instance by installing the software that you need.
3. Create an HVM AMI from your running cluster instance.
4. Create a placement group.
5. Launch cluster instances into the placement group.
6. (Optional) Use AWS CloudFormation to automate these tasks for you. See [AWS CloudFormation Templates for Cluster Instances \(p. 113\)](#).

The next sections show you how to complete these steps using the AWS Management Console, the command line tools, or the API.

AWS Management Console

To launch a cluster instance

1. Launch a cluster instance just as you would launch any other type of instance.
For general instructions, see [Launching an Instance from an AMI \(p. 284\)](#).
2. When prompted to choose an AMI, select the Cluster Compute Amazon Linux AMI.
3. When prompted to edit the instance details, select the instance type that meets your needs. Notice that you can only select high I/O and cluster instance types with these AMIs.

To customize your cluster instance

1. Customize your cluster instance just as you would any other type of instance. First, connect to your instance. For instructions, see [Connecting to Amazon EC2 Instances \(p. 293\)](#).
2. Install software and applications on the instance, copy data, or attach additional EBS volumes. After you've configured your instance to meet your needs, you're ready to create an AMI.

To create an HVM AMI from your running cluster instance

1. Create an AMI from your cluster instance just as you would create an AMI from any other type of instance. For instructions that use the AWS Management Console, see [Creating Amazon EBS-Backed AMIs Using the Console \(p. 28\)](#).
2. After you've created your AMI, you can terminate the instance that served as the basis for your AMI. Now you're ready to launch instances into a placement group using this AMI that you've created.

To create a placement group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation bar, select the region for the placement group.
3. In the navigation pane, click **Placement Groups**.
4. Click **Create Placement Group**.
5. In the **Create Placement group** dialog box, provide a unique name for the placement group, and then click **Yes, Create**.

When the status of the placement group is available, you're ready to launch cluster instances into the placement group.

To launch cluster instances into your placement group

1. Launch an instance using the Classic Wizard.
For general instructions, see [Launching an Instance from an AMI \(p. 284\)](#).
2. On the **CHOOSE AN AMI** page, click the **My AMIs** tab, and select the HVM AMI that you created.
3. When prompted by the Request Instances Wizard, select the placement group that you created from the **Placement Group** list.

Command Line Interface

To launch a cluster instance

1. Launch a cluster instance using the `ec2-run-instances` command, just as you would launch any other type of instance.
Be sure to specify the Cluster Compute Amazon Linux AMI. This is an Amazon EBS-backed HVM AMI designed for cluster computing. Also specify one of the cluster instance types.
2. The following example launches a `cc2.8xlarge` instance.

```
PROMPT> ec2-run-instances ami-xxxxxxxx --instance-type cc2.8xlarge
```

To customize your cluster instance

1. Customize your cluster instance just as you would any other type of instance. First, connect to your instance. For instructions, see [Connecting to Amazon EC2 Instances \(p. 293\)](#).
2. Install software and applications on the instance, copy data, or attach additional EBS volumes. After you've configured your instance to meet your needs, you're ready to create an AMI.

To create an HVM AMI from your running cluster instance

1. Use the `ec2-create-image` command to create your own image from your customized instance.

```
PROMPT> ec2-create-image -n your_image_name instance_id
```

For example:

```
PROMPT> ec2-create-image -n "My_Custom_HVM_AMI" i-1a2b3c4d
```

Amazon EC2 begins creating the image and returns the new image's ID. For example:

```
IMAGE ami-5f4d3a2b
```

2. To check whether the AMI is ready, use the following command.

```
PROMPT> ec2-describe-images -o self
```

Amazon EC2 returns information about the AMI.

To create a placement group

1. Use the `ec2-create-placement-group` command. You must name the placement group and set the `-s` option to `cluster`, as shown in the following example.

```
PROMPT> ec2-create-placement-group XYZ-cluster -s cluster
```

The following is example output.

```
PLACEMENTGROUP XYZ_cluster cluster available
```

2. Record the name of the placement group. You'll need this name when you launch your cluster instances into this group.
3. Use the `ec2-describe-placement-groups` command to get details about your cluster placement groups.

If you do not specify the name of the cluster placement group, details about all the placement groups in your account are returned.

```
PROMPT> ec2-describe-placement-groups
```

The following is example output.

```
PLACEMENTGROUP XYZ-cluster cluster available
PLACEMENTGROUP ABC-cluster cluster available
```

To launch cluster instances into your placement group

- Use the `ec2-run-instances` command with the name of the placement group.

The following example launches 10 cc2.8xlarge instances into the XYZ-cluster placement group.

```
PROMPT> ec2-run-instances ami-5f4d3a2b --instance-type cc2.8xlarge -n 10 -  
-placement-group XYZ-cluster
```

The following is example output.

```
RESERVATION      r-064bcd6d      999988887777    default  
INSTANCE i-89fea78c ... cc2.8xlarge ... us-east-1a ... XYZ-cluster hvm  
...
```

API

To launch a cluster instance

1. Launch a cluster instance using the `RunInstances` action, just as you would launch any other type of instance.

Be sure to specify the Cluster Compute Amazon Linux AMI. This is an Amazon EBS-backed HVM AMI designed for cluster computing. Also specify one of the cluster instance types.

2. The following example launches a cc2.8xlarge instance.

```
https://ec2.amazonaws.com/?Action=RunInstances  
&ImageId=ami-xxxxxxxx  
&InstanceType=cc2.8xlarge  
&MaxCount=1  
&MinCount=1  
&AUTHPARAMS
```

To customize your cluster instance

1. Customize your cluster instance, just as you would any other type of instance. First, connect to your instance. For instructions, see [Connecting to Amazon EC2 Instances \(p. 293\)](#).
2. Install software and applications on the instance, copy data, or attach additional EBS volumes. After you've configured your instance to meet your needs, you're ready to create an AMI.

To create an HVM AMI from your running cluster instance

- Construct a Query request, as shown in the following example.

```
https://ec2.amazonaws.com/  
?Action/CreateImage  
&InstanceId=i-1a2b3c4d  
&Name=My_Custom_HVM_AMI  
&AUTHPARAMS...
```

To create a placement group

1. Construct a Query request, as shown in the following example. You must name the placement group and specify the strategy as `cluster`.

```
Action=CreatePlacementGroup  
&GroupName=XYZ-cluster  
&Strategy=cluster  
&AUTHPARAMS...
```

The following is an example response.

```
<CreatePlacementGroupResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">  
    <requestID>d4904fd9-82c2-4ea5-adfe-a9cc3EXAMPLE</requestId>  
    <return>true</return>  
</CreatePlacementGroupResponse>
```

2. To get information about your cluster placement groups, construct a Query request similar to the following.

```
Action=DescribePlacementGroups  
&GroupName.1=XYZ-cluster  
&AUTHPARAMS...
```

The following is an example response.

```
<DescribePlacementGroupsResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">  
    <requestId>d4904fd9-82c2-4ea5-adfe-a9cc3EXAMPLE</requestId>  
    <placementGroupSet>  
        <item>  
            <groupName>XYZ-cluster</groupName>  
            <strategy>cluster</strategy>  
            <state>available</state>  
        </item>  
    </placementGroupSet>  
</DescribePlacementGroupsResponse>
```

To launch cluster instances into your placement group

- Construct a Query request similar to the following example. This example launches 10 cc2.8xlarge instances in the XYZ-cluster placement group.

```
Action=RunInstances  
&InstanceType=cc2.8xlarge  
&Placement.GroupName=XYZ-cluster  
&MinCount=10  
&MaxCount=10  
&AUTHPARAMS...
```

The following is an example response.

```
<RunInstancesResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">...
<placement>
<availabilityZone>us-east-1a</availabilityZone>
<groupName>XYZ-cluster</groupName>
</placement>
...
<virtualizationType>hvm</virtualizationType>
</RunInstancesResponse>
```

How to Delete a Placement Group

You can delete a placement group if your cluster instance requirements change or if you no longer need the placement group.

AWS Management Console

Before you delete your placement group, first you need to terminate all running instances in the group.

To terminate an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Instances**.
3. Right-click the instance, and then click **Terminate**.
4. When prompted for confirmation, click **Yes, Terminate**.

After you've terminated the cluster instances in a placement group, use the following steps to delete the placement group.

To delete a placement group

1. In the navigation pane, click **Placement Groups**.
2. Select the placement group, and click **Delete**.
3. When prompted for confirmation, click **Yes, Delete**.

Command Line Interface

Before you delete your placement group, first you need to terminate all running instances in the group.

To terminate an instance

- Use the `ec2-terminate-instances` command. This example command terminates the instance with the instance ID i-10a64379.

```
ec2-terminate instances i-10a64379
```

The following is an example of the output for this command.

```
INSTANCE i-10a64379 running shutting-down
```

After you've terminated the cluster instances in a placement group, use the following steps to delete the placement group.

To delete a placement group

- Use the `ec2-delete-placement-group` command. This example command deletes the placement group named XYZ-cluster.

```
PROMPT> ec2-delete-placement-group XYZ-cluster
```

The following is an example of the output for this command.

```
PLACEMENTGROUP  XYZ-cluster  deleted
```

API

Before you delete your placement group, first you need to terminate all running instances in the group.

To terminate an instance

- Construct a Query request similar to the following example.

```
https://ec2.amazonaws.com/?Action=TerminateInstances
&InstanceId.1=i-10a64379
&AUTHPARAMS
```

The following is an example response.

```
<TerminateInstancesResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
  <requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId>
  <instancesSet>
    <item>
      <instanceId>i-10a64379</instanceId>
      <currentState>
        <code>32</code>
        <name>shutting-down</name>
      </currentState>
      <previousState>
        <code>16</code>
        <name>running</name>
      </previousState>
    </item>
  </instancesSet>
</TerminateInstancesResponse>
```

After you've terminated the cluster instances in a placement group, use the following steps to delete the placement group.

To delete a placement group

- Construct a Query request similar to the following example.

```
Action=DeletePlacementGroup
&GroupName=XYZ-cluster
&AUTHPARAMS...
```

The following is an example response.

```
<DeletePlacementGroupResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
  <requestID>d4904fd9-82c2-4ea5-adfe-a9cc3EXAMPLE</requestId>
  <return>true</return>
</DeletePlacementGroupResponse>
```

AWS CloudFormation Templates for Cluster Instances

To make it easy for you to launch a cluster, you can use AWS CloudFormation templates that provision a two-node cluster with NFS storage, monitoring and a mounted 200 GB data set.

Each template performs the following tasks:

1. Creates a new controller instance.
2. Bootstraps MIT StarCluster. For more information about MIT StarCluster, see <http://star.mit.edu/cluster/>.
3. Generates a new cluster configuration.
4. Creates a new data volume.
5. Creates a new cluster master, worker instance, and storage.
6. Configures the resources to process compute jobs using Oracle Grid Engine.

You can use these templates to create a stack, log into the controller image, and use the cluster master to submit jobs or configure your cluster.

Here is where you can download the templates from AWS:

- Download the template for a `cc2.8xlarge` cluster from
<https://s3.amazonaws.com/cloudformation-templates-us-east-1/cc2-cluster.json>
- Download the template for a `cc1.4xlarge` cluster from
<https://s3.amazonaws.com/cloudformation-templates-us-east-1/cc1-cluster.json>

For more information about using AWS CloudFormation, see the [AWS CloudFormation User Guide](#).

Using the NVIDIA Driver on a Cluster GPU Instance

We provide reference AMIs for cluster instances that already have the NVIDIA driver installed. Amazon provides updated and compatible builds of the NVIDIA kernel drivers for each new official kernel upgrade. If you decide to use a different NVIDIA driver version than the one Amazon provides, or decide to use a kernel that's not an official Amazon build, then you first need to uninstall the Amazon-provided NVIDIA packages from your system to avoid conflicts with the versions of the drivers you are trying to install.

Use this command to uninstall Amazon-provided NVIDIA packages:

```
[root@ip-xxx ~]# sudo yum erase nvidia cudatoolkit
```

Important

The Amazon-provided CUDA toolkit package has dependencies on the NVIDIA drivers. Uninstalling the NVIDIA packages erases the CUDA toolkit. You will need to reinstall the CUDA toolkit after installing the NVIDIA driver.

Manual Installation of the NVIDIA Driver

A cluster GPU instance must have the NVIDIA Tesla M2050 GPU driver. The NVIDIA driver you install needs to be compiled against the kernel you intend to run on your instance.

Tip

You can download NVIDIA drivers at <http://www.nvidia.com/Download/Find.aspx>. Look for the Tesla M-Class M2050 driver for Linux 64-bit systems.

To install the driver for an Amazon Linux AMI

1. Make sure the `kernel-devel` package is installed and matches the version of the kernel you are currently running.

```
[root@ip-xxx ~]# yum install kernel-devel- uname -r
```

2. Install the NVIDIA driver.

```
[root@ip-xxx ~]# /root/NVIDIA-Linux-x86_64_260.19.12.run
```

3. Reboot the instance.
4. Confirm that the driver is functional.

```
[root@ip-xxx ~]# /usr/bin/nvidia-smi -q -a
```

The response lists the installed NVIDIA driver version and details about the GPUs.

5. Confirm that the GPUs are exposed.

```
[root@ip-xxx ~]# ls /dev/*nv*
/dev/nvidia0 /dev/nvidia1 /dev/nvidiactl /dev/nvram
```

If a newer version of the NVIDIA driver is available, you might want to update your instance to use it. To update your instance, follow the steps in the preceding section for installing the driver, and run the self-install script from the newly downloaded driver version.

To install the CUDA toolkit package, download the package from the NVIDIA website, and run the self-install script as outlined in step 2 of the preceding section for installing the driver.

Cluster Instance Hardware Specifications

Hardware specifications for the `ccl.4xlarge` cluster compute instance type, `cgl.4xlarge` cluster GPU instance type, `crl.8xlarge` high-memory cluster instance type, and `cc2.8xlarge` cluster compute instance type are described in the following table.

Item	cc1.4xlarge cluster compute instance type or cg1.4xlarge cluster GPU instance type	cr1.8xlarge high-memory cluster instance type	cc2.8xlarge cluster compute instance type
Processor	33.5 EC2 Compute Units (2 x Intel Xeon X5570, quad-core)	88 EC2 Compute Units (2 x Intel Xeon E5-2670, eight-core)	88 EC2 Compute Units (2 x Intel Xeon E5-2670, eight-core)
Hyperthreading	8 physical cores with 16 threads	16 physical cores with 32 threads	16 physical cores with 32 threads
Memory	23 GiB Note: The cluster GPU instances can use up to 22 GiB, with 1 GiB reserved for GPU operation. The 22 GiB doesn't include the GPU on-board memory, which is 3 GiB per GPU for the NVIDIA Tesla M2050 GPU.	244 GiB	60.5 GiB
Platform	64-bit	64-bit	64-bit
Network	10 Gbps Ethernet	10 Gbps Ethernet	10 Gbps Ethernet
Local Storage	1680 GB instance 64-bit storage (2 x 840 GB)	240 GB instance 64-bit storage (2 x 120 GB SSD)	3360 GB instance 64-bit storage (4 x 840 GB)
GPUs (cluster GPU instances only)	Two NVIDIA Tesla M2050 GPUs Note: The instance must have the corresponding NVIDIA driver installed; the reference AMI we provide includes the driver. If you need to download and install the driver, see Manual Installation of the NVIDIA Driver (p. 114) .	N/A	N/A

When determining the number of cluster instances you can launch, be aware that you're limited to a maximum of 20 instances, of which two can be cluster GPU instances. If you need more instances, use the [Request to Increase Amazon EC2 Instance Limit](#) form.

For more information, see [Instance Families and Types \(p. 87\)](#).

Regions and Availability Zones

Amazon EC2 is hosted in multiple locations world-wide. These locations are composed of regions and Availability Zones. Each *region* is a separate geographic area. Each region has multiple, isolated locations known as *Availability Zones*. Amazon EC2 provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across regions unless you do so specifically.

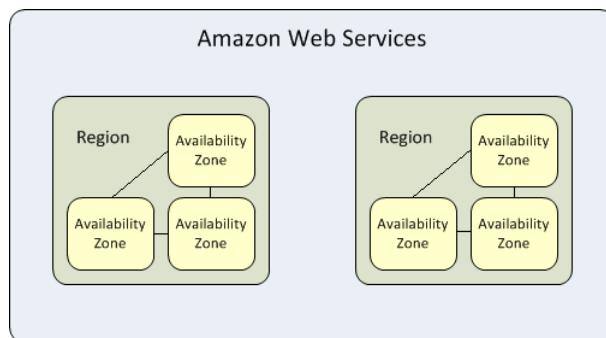
Amazon operates state-of-the-art, highly-available data centers. Although rare, failures can occur that affect the availability of instances that are in the same location. If you host all your instances in a single location that is affected by such a failure, none of your instances would be available.

Topics

- [Region and Availability Zone Concepts \(p. 116\)](#)
- [Describing Your Regions and Availability Zones \(p. 117\)](#)
- [Specifying the Region for a Resource \(p. 118\)](#)
- [Launching Instances in an Availability Zone \(p. 119\)](#)
- [API and Command Overview \(p. 120\)](#)

Region and Availability Zone Concepts

Each region is completely independent. Each Availability Zone is isolated, but the Availability Zones in a region are connected through low-latency links. The following diagram illustrates the relationship between regions and Availability Zones.



Some Amazon EC2 resources are tied to a region and others are tied to an Availability Zone. For example, an instance is tied to an Availability Zone and an Amazon EBS volume is tied to a region. For more information, see [Resource Locations \(p. 534\)](#).

Regions

Each Amazon EC2 region is designed to be completely isolated from the other Amazon EC2 regions. This achieves the greatest possible fault tolerance and stability.

Amazon EC2 provides multiple regions so that 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.

When you view your resources, you'll only see the resources tied to the region you've specified. This is because regions are isolated from each other, and we don't replicate resources across regions automatically.

When you work with an instance using the command line interface or API actions, you must specify its regional endpoint. For more information about the regions and endpoints for Amazon EC2, see [Regions and Endpoints](#) in the *Amazon Web Services General Reference*.

When you launch an instance, you must select an AMI that's in the same region. If the AMI is in another region, you can copy the AMI to the region you're using. For more information, see [Copying AMIs \(p. 67\)](#).

All communications between regions is across the public Internet. Therefore, you should use the appropriate encryption methods to protect your data. Data transfer between regions is charged at the Internet data transfer rate for both the sending and the receiving instance. For more information, see [Amazon EC2 Pricing - Data Transfer](#).

Availability Zones

You can list the Availability Zones that are available to your account. For more information, see [Describing Your Regions and Availability Zones \(p. 117\)](#).

When you launch an instance, you can select an Availability Zone or let us chose one for you. If you distribute your instances across multiple Availability Zones and one instance fails, you can design your application so that an instance in another Availability Zone can handle requests.

You can also use Elastic IP addresses to mask the failure of an instance in one Availability Zone by rapidly remapping the address to an instance in another Availability Zone. For more information, see [Elastic IP Addresses \(EIP\) \(p. 438\)](#).

To ensure that resources are distributed across the Availability Zones for a region, we independently map Availability Zones to identifiers for each account. For example, your Availability Zone `us-east-1a` might not be the same location as `us-east-1a` for another account. Note that there's no way for you to coordinate Availability Zones between accounts.

As Availability Zones grow over time, our ability to expand them can become constrained. If this happens, we might restrict you from launching an instance in a constrained Availability Zone unless you already have an instance in that Availability Zone. Eventually, we might also remove the constrained Availability Zone from the list of Availability Zones for new customers. Therefore, your account might have a different number of available Availability Zones in a region than another account.

Describing Your Regions and Availability Zones

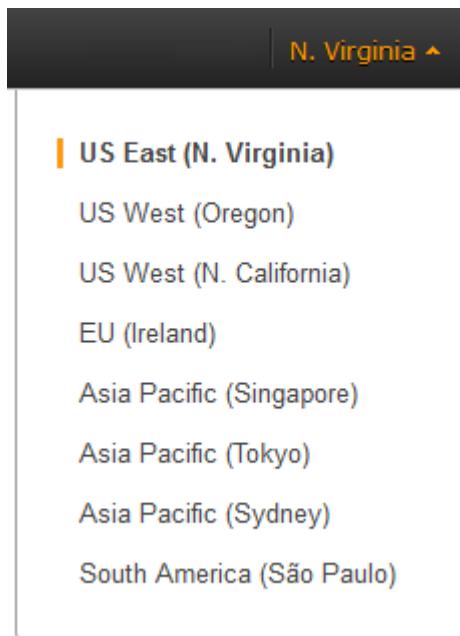
You can use the AWS Management Console or the Amazon EC2 command line interface to determine which regions and Availability Zones are available for your use.

```
PROMPT> > ec2-describe-availability-zones --region us-east-1
AVAILABILITYZONE us-east-1a available us-east-1
AVAILABILITYZONE us-east-1b available us-east-1
AVAILABILITYZONE us-east-1c available us-east-1
AVAILABILITYZONE us-east-1d available us-east-1
```

AWS Management Console

To find your regions and Availability Zones

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, view the options in the region selector.



3. After you select a region, you can view your Availability Zones within that region when you launch an instance or create an Amazon EBS volume.
 - a. In the navigation pane, click **Volumes**.
 - b. View the options in the **Availability Zones** list.
 - c. When you are finished, click **Cancel**.

Command Line Interface

Use the following command to describe your regions.

```
PROMPT> ec2-describe-regions
REGION us-east-1 ec2.us-east-1.amazonaws.com
REGION ap-northeast-1 ec2.ap-northeast-1.amazonaws.com
REGION ap-southeast-1 ec2.ap-southeast-1.amazonaws.com
..
```

Use the following command to describe the Availability Zones within the `us-east-1` region.

```
PROMPT> > ec2-describe-availability-zones --region us-east-1
AVAILABILITYZONE us-east-1a available us-east-1
AVAILABILITYZONE us-east-1b available us-east-1
AVAILABILITYZONE us-east-1c available us-east-1
AVAILABILITYZONE us-east-1d available us-east-1
```

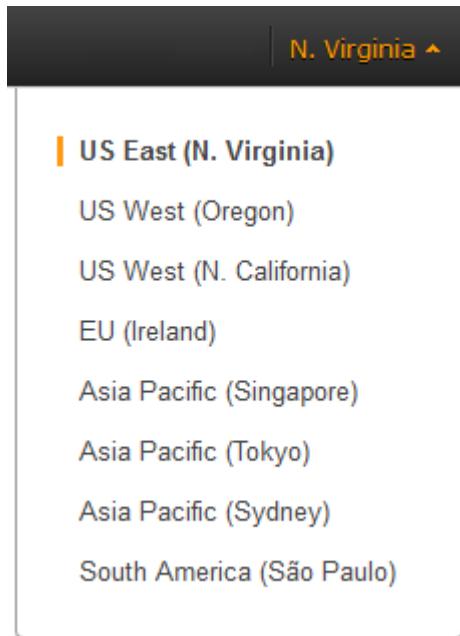
Specifying the Region for a Resource

Every time you create an Amazon EC2 resource, you can specify the region for the resource. This section explains how to specify the region for a resource.

AWS Management Console

To specify the region for a resource

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Use the region selector in the navigation bar.



Command Line Interface

To specify the region to use for all commands, set the value of the `EC2_URL` environment variable to the regional endpoint. For example, `https://ec2.us-west-1.amazonaws.com`.

Alternatively, you can use the `--region` or `-U` command line option with each individual command. For example, `--region us-west-1` or `-U https://ec2.us-west-1.amazonaws.com`.

For more information about the endpoints for Amazon EC2, see [Amazon Elastic Compute Cloud Endpoints](#).

Launching Instances in an Availability Zone

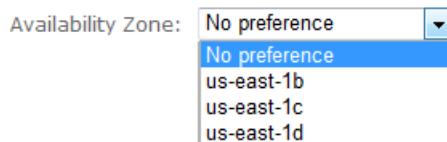
When you launch an instance, select a region that puts your instances closer to specific customers, or meets the legal or other requirements you have. By launching your instances in separate Availability Zones, you can protect your applications from the failure of a single location.

When you launch an instance, you can optionally specify an Availability Zone in the region that you are using. If you do not specify an Availability Zone, we select one for you. When you launch your initial instances, we recommend that you accept the default Availability Zone, because this enables us to select the best Availability Zone for you based on system health and available capacity. If you launch additional instances, only specify an Availability Zone if your new instances must be close to, or separated from, your running instances.

AWS Management Console

To specify an Availability Zone for your instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the dashboard, click **Launch Instance**.
3. Follow the directions for the wizard to launch the instance. You can select one of the Availability Zone options from the list, or select **No preference** to enable us to select the best Availability Zone for you.



Command Line Interface

To specify an Availability Zone for your instance, use the `--availability-zone` option with the `ec2-run-instances` command.

```
PROMPT> ec2-run-instances ami_id --availability-zone zone
```

API and Command Overview

The following table summarizes the available commands and corresponding API actions for regions and Availability Zones.

Description	Command and API Action
Describes the Availability Zones that are available to you.	ec2-describe-availability-zones DescribeAvailabilityZones
Describes the regions that are available to you.	ec2-describe-regions DescribeRegions

Supported Platforms

Amazon EC2 supports the following platforms. Your AWS account is capable of launching instances either into both platforms or only into EC2-VPC, on a region by region basis.

Platform	Introduced In	Description
EC2-Classic	The original release of Amazon EC2	Your instances run in a single, flat network that you share with other customers.
EC2-VPC	The original release of Amazon VPC	Your instances run in a virtual private cloud (VPC) that's logically isolated to your AWS account.

Supported Platforms in the Amazon EC2 Console

The Amazon EC2 console indicates which platforms you can launch instances into for the selected region, and whether you have a default VPC in that region.

Verify that the region you'll use is selected in the navigation bar. On the Amazon EC2 console dashboard, look for **Supported Platforms** under **Account Attributes**. If there are two values, EC2-Classic and EC2-VPC, you can launch instances into either platform. If there is one value, EC2-VPC, you can launch instances only into EC2-VPC.

If you can launch instances only into EC2-VPC, we create a default VPC for you. Then, when you launch an instance, we launch it into your default VPC, unless you create a nondefault VPC and specify it when you launch the instance.

EC2-VPC

The dashboard displays the following under **Account Attributes** to indicate that the account supports only the EC2-VPC platform, and has a default VPC with the identifier `vpc-1a2b3c4d`.

Supported Platforms
EC2-VPC

Default VPC
`vpc-1a2b3c4d`

If your account supports only EC2-VPC, you can select a subnet from the **Subnet** list when you launch an instance using the Classic wizard.

Launch into:

Subnet:	No Preference (default subnet in any AZ)	* denotes default subnet
---------	------------------------------------------	--------------------------

EC2-Classic, EC2-VPC

The dashboard displays the following under **Account Attributes** to indicate that the account supports both the EC2-Classic and EC2-VPC platforms.

Supported Platforms
EC2-Classic
EC2-VPC

If your account supports EC2-Classic and EC2-VPC, you can use the **EC2-Classic** tab to launch an instance into EC2-Classic using the Classic wizard.

Launch into:

<input checked="" type="radio"/> EC2-Classic	<input type="radio"/> EC2-VPC
Availability Zone:	No Preference

If your account supports EC2-Classic and EC2-VPC, you can use the **EC2-VPC** tab to select a subnet when you launch an instance into a VPC using the Classic wizard.

Launch into:

<input type="radio"/> EC2-Classic	<input checked="" type="radio"/> EC2-VPC
Subnet:	subnet-967673fd (10.0.0.0/24) us-east-1b
249 available IP addresses	

Related Topic

For more information about how you can tell which platforms you can launch instances into, see [Detecting Your Supported Platforms](#) in the *Amazon Virtual Private Cloud User Guide*.

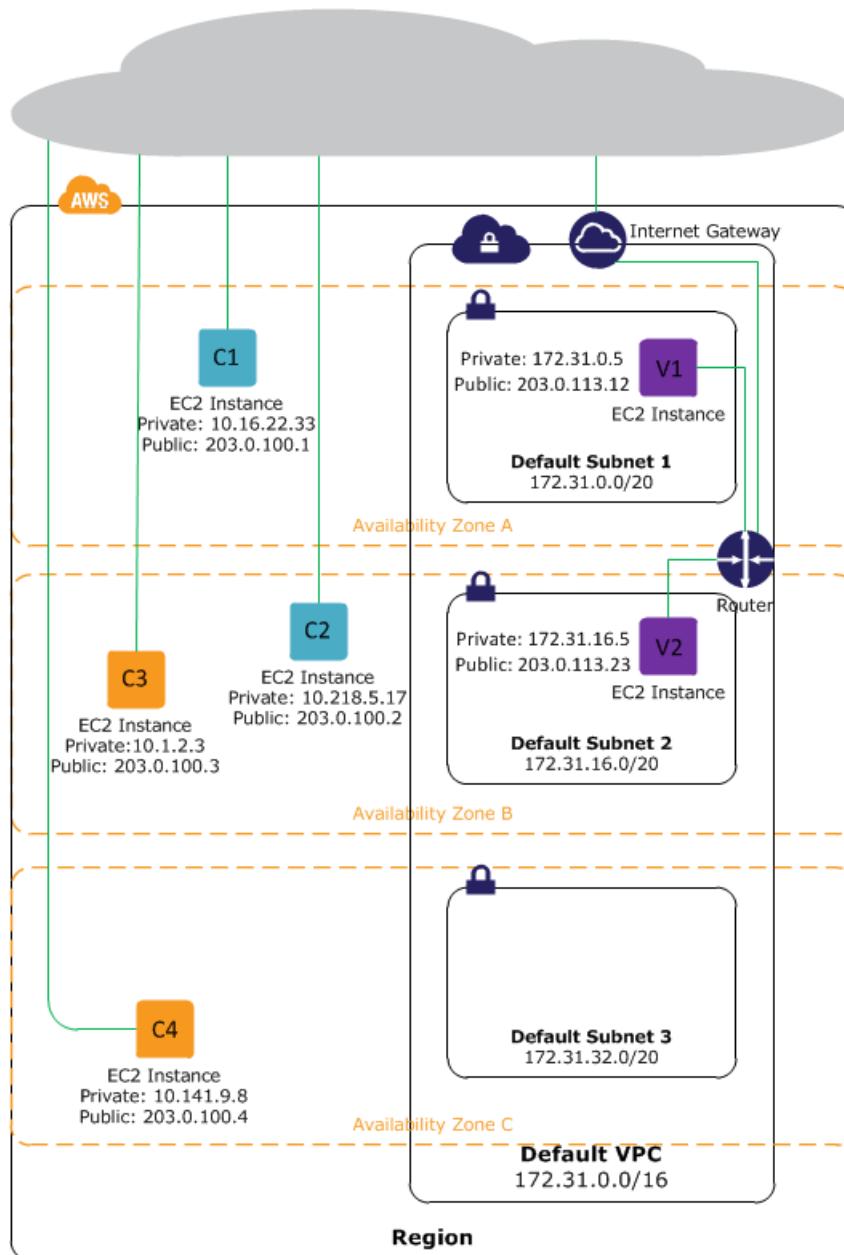
Differences Between Instances in EC2-Classic and EC2-VPC

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

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

The following diagram shows instances in each platform. Note the following:

- Instances C1, C2, C3, and C4 are in the EC2-Classic platform. C1 and C2 were launched by one account, and C3 and C4 were launched by a different account. These instances can communicate with each other, can access the Internet directly, and can access other Amazon Web Services such as Amazon Simple Storage Service (Amazon S3).
- Instances V1 and V2 are in different subnets in the same VPC in the EC2-VPC platform. They were launched by the account that owns the VPC; no other account can launch instances in this VPC. These instances can communicate with each other and can access the following through the Internet gateway: instances in EC2-Classic, other Amazon Web Services (such as Amazon S3), and the Internet.



For more information about the differences between EC2-Classic and EC2-VPC, see [Amazon EC2 and Amazon Virtual Private Cloud \(VPC\) \(p. 428\)](#).

Amazon EC2 Root Device Volume

When you launch an Amazon EC2 instance, the *root device volume* contains the image used to boot the instance. When we introduced Amazon EC2, all AMIs were backed by Amazon EC2 instance store, which means the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3. After we introduced Amazon EBS, we introduced AMIs that are backed by Amazon EBS. This means that the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot. You can choose between AMIs based by Amazon EC2 instance store and AMIs backed by Amazon EBS. We recommend that you use AMIs backed by Amazon EBS, because they launch faster and use persistent storage.

Topics

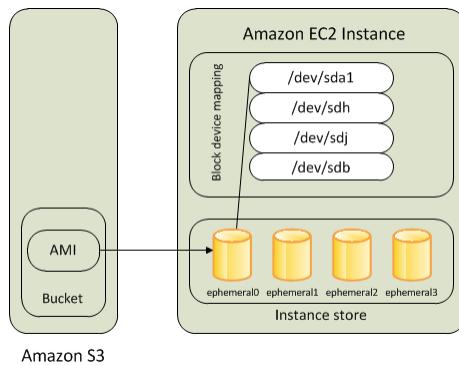
- [Root Device Storage Concepts \(p. 124\)](#)
- [Root Device Storage Usage Scenarios \(p. 125\)](#)
- [Changing the Root Device Volume to Persist \(p. 128\)](#)

Root Device Storage Concepts

You can launch an instance from one of two types of AMIs: an Amazon EC2 instance store-backed AMI or an Amazon EBS-backed AMI. The description of an AMI includes type of AMI it is; you'll see the root device referred to in some places as either `ebs` (for Amazon EBS-backed) or `instance store` (for Amazon EC2 instance store-backed). This is important because there are significant differences between what you can do with each type of AMI. For more information about these differences, see [Storage for the Root Device \(p. 19\)](#).

Instance Store-backed Instances

Instances that use instance stores for the root device automatically have instance store volumes available, with one serving as the root device volume. When an instance is launched, the image that is used to boot the instance is copied to the root volume (typically `sda1`). Any data on the instance store volumes persists as long as the instance is running and is deleted when the instance is stopped, terminated, or fails (such as if an underlying drive fails).

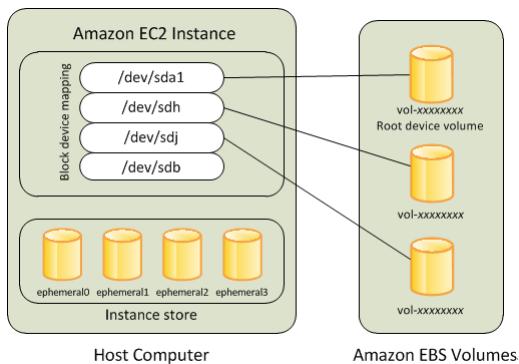


After an instance store-backed instance fails or terminates, it cannot be restored. If you plan to use Amazon EC2 instance store-backed instances, we highly recommend that you distribute the data on your instance stores across multiple Availability Zones. You should also back up the data on your instance store volumes to Amazon S3 on a regular basis.

For more information, see [Amazon EC2 Instance Store \(p. 513\)](#).

Amazon EBS-backed Instances

Instances that use Amazon EBS for the root device automatically have an Amazon EBS volume attached. When you launch an Amazon EBS-backed instance, we create an Amazon EBS volume for each EBS snapshot referenced by the AMI you use. You can optionally use other Amazon EBS volumes or instance store volumes.



An Amazon EBS-backed instance can be stopped and later restarted without affecting data stored in the attached volumes. There are various instance- and volume-related tasks you can do when an Amazon EBS-backed instance is in a stopped state. For example, you can modify the properties of the instance, you can change the size of your instance or update the kernel it is using, or you can attach your root volume to a different running instance for debugging or any other purpose.

By default, the root device volume and the other Amazon EBS volumes attached when you launch an Amazon EBS-backed instance are automatically deleted when the instance terminates. For information about how to change this behavior when you launch an instance, see [Changing the Root Device Volume to Persist \(p. 128\)](#).

By default, any Amazon EBS volumes that you attach to a running instance are detached with their data intact when the instance terminates. You can attach a detached volume to any running instance.

If an Amazon EBS-backed instance fails, you can restore your session by following one of these methods:

- Stop and then start again.
- Automatically snapshot all relevant volumes and create a new AMI. For more information, see [Creating Amazon EBS-Backed Linux AMIs \(p. 31\)](#).
- Attach the volume to the new instance by following these steps:
 1. Create a snapshot of the root volume.
 2. Register a new AMI using the snapshot.
 3. Launch a new instance from the new AMI.
 4. Detach the remaining Amazon EBS volumes from the old instance.
 5. Reattach the Amazon EBS volumes to the new instance.

We recommend using either the first or the second method for failed instances with normal volume size and the third method for failed instances with large volumes.

Root Device Storage Usage Scenarios

You can implement Amazon EBS-backed AMIs by creating a set of snapshots and registering an AMI that uses those snapshots. The AMI publisher controls the default size of the root device through the size of the snapshot. The default size can be increased up to 1TiB to accommodate the requirements of the application either at the time you register the Amazon EBS-backed AMI or while you launch the Amazon EBS-backed instance.

Launching an Amazon EBS-backed instance with increased root device storage disk size

- | | |
|---|---------------------------------------------------------------|
| 1 | Select an Amazon EBS-backed AMI to launch your instance from. |
|---|---------------------------------------------------------------|

2	Check the root device size and note the AMI ID.
3	Using the command line interface, launch the instance by specifying the AMI ID and the mapping of the root device with the increased size.
4	Connect to the instance.
5	Check the size of the root device on the instance. The increased size of the root device is not apparent yet. This is because the file system does not recognize the increased size on the root device.
6	Resize the file system.
7	Check the size of the root device.
8	The root device of the newly launched instance now shows the increased size.

Note

You cannot decrease the size of your root device to less than the size of the AMI. To decrease the size of your root device, create your own AMI with the desired size for the root device and then launch an instance from that AMI.

Increasing the size of the root device on an Amazon EBS-backed running instance

1	Get the Amazon EBS volume ID and the Availability Zone of a running instance for which you want to increase the root storage size.
2	Stop the instance.
3	Detach the original volume from the instance.
4	Create a snapshot of the detached volume.
5	Create a new volume from the snapshot by specifying a larger size.
6	Attach the new volume to the stopped instance.
7	Start the instance and get the new IP address/hostname.
8	Connect to the instance using the new IP address/hostname.
9	Resize the root file system to the extent of the new Amazon EBS volume.
10	Check the size of the root device. The root device now shows the increased size.
11	[Optional] Delete the old Amazon EBS volume, if you no longer need it.

The following list describes the tasks for creating a snapshot of the root device of an Amazon EC2 instance store-backed instance. The snapshot is created using an Amazon EBS volume. You can use this snapshot to create a new Amazon EBS-backed AMI or to launch another instance.

Creating a snapshot of the root device of an Amazon EC2 instance store-backed instance

1	Launch an instance from an Amazon EC2 instance store-backed AMI.
---	------------------------------------------------------------------

2	Create a 10GiB Amazon EBS volume in the same Availability Zone as that of your newly launched instance.
	Note Use this volume to create a snapshot of the root partition of an Amazon EC2 instance store-backed AMI. The resulting snapshot is the same size as the root partition; the maximum size of the root partition in an Amazon EC2 instance store-backed AMI is 10GiB.
3	Attach the volume to the running instance using either the AWS Management Console or the command line tools.
4	Format the volume with a file system.
5	[For Linux Users] Create a directory and then mount the volume on the newly-created directory.
6	Copy the data on the root storage device to the newly-attached volume.
7	Unmount and detach the volume from the instance.
8	Create a snapshot of the volume.

Amazon EC2 instance store-backed AMIs are limited to 10GiB storage for the root device. If you require additional storage on your root device, you must first convert the Amazon EC2 instance store-backed AMI to an Amazon EBS-backed AMI and then launch an Amazon EBS-backed instance with increased root storage.

Note

This conversion procedure works with a Linux AMI, but step 6 fails with a Windows AMI.

**Converting an Amazon EC2 instance store-backed AMI to an Amazon EBS-backed AMI
(Linux only)**

1	Launch an instance from an Amazon EC2 instance store-backed AMI.
2	Create a 10GiB Amazon EBS volume in the same Availability Zone as that of your newly-launched instance. Use this volume to create a snapshot of the root partition of the Amazon EC2 instance store-backed AMI. The resulting snapshot is the same size as the root partition; the maximum size of the root partition in an Amazon EC2 instance store-backed AMI is 10GiB.
3	Attach the volume to the running instance using either the AWS Management Console or the command line interface.
4	Format the volume with a file system.
5	Create a directory and then mount the volume on the newly-created directory.
6	Copy the data on the root storage device to the newly-attached volume.
7	Unmount and detach the volume from the instance.
8	Create a snapshot of the volume.
9	Register the snapshot of the volume as an AMI.

Changing the Root Device Volume to Persist

By default, the root device volume for an AMI backed by Amazon EBS is deleted when the instance terminates. To change the default behavior, set the `DeleteOnTermination` flag to `false` in the instance's block device mapping.

AWS Management Console

To change the root device volume to persist when you launch an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the Amazon EC2 console dashboard, click **Launch Instance**.
3. On the **Create a New Instance** page, click **Classic Wizard**.
4. On the **CHOOSE AN AMI** page, choose the AMI to use and click **Select**.
5. Follow the wizard to complete the first and second **INSTANCE DETAIL** pages.
6. On the third **INSTANCE DETAIL** page, under **Storage Device Configuration**, click **Edit**.
7. Deselect **Delete On Termination** on the **Root Volume** tab, click **Save**, and then click **Continue**.

The screenshot shows the 'Request Instances Wizard' interface. The 'INSTANCE DETAILS' tab is selected. Under 'Storage Device Configuration', it says 'Number of Instances: 1' and 'Availability Zone: No Preference'. A section titled 'Storage Device Configuration' contains a note: 'Your instance will be launched with the following storage device settings. Edit these settings to add EBS volumes, instance store volumes, or edit the settings of the root volume.' Below this, there are three radio buttons: 'Root Volume' (selected), 'EBS Volumes', and 'Instance Store Volumes'. A note below says 'Optionally, edit the root volume of your instance and then click Save.' It shows 'Volume Size: 8 GiB', 'Volume Type: Standard', 'IOPS: 100', 'Device: /dev/sda1', and 'Delete on Termination: checked'. A 'Save' button is shown. A table below lists storage devices:

Type	Device	Snapshot ID	Size	Volume Type	IOPS	Delete on Termination
Root	/dev/sda1	snap-495fcc0e	8	standard		true

At the bottom, it says '0 EBS Volumes 0 Ephemerals'. Navigation buttons 'Back' and 'Continue' are at the bottom right.

8. Complete the remaining wizard pages, and then click **Launch Instance**.

You can verify the setting by viewing details for the root device volume on the instance's details pane in the AWS Management Console. Under **Block Devices**, click the entry for the root device volume. By default, **Delete on termination** is Yes. If you change the default behavior, **Delete on termination** is No.

sda1

EBS ID:	vol-7f110939
Root device type:	ebs
Attachment time:	2013-04-09T19:32:25.000Z
Block device status:	attached
Delete on termination:	No
Snapshot ID:	snap-495fcc0e

Command Line Interface

Use the `ec2-run-instances` command, and include a block device mapping that sets the `deleteOnTermination` flag for the root device to `false`. Include the `-v` option to run the command in verbose mode.

```
PROMPT> ec2-run-instances ami_id -b root_device_name=:false other parameters... -v
```

The root device is typically `/dev/sda1`, or `xvda` (for Windows). The following is an example.

```
PROMPT> ec2-run-instances ami-1a2b3c4d -b /dev/sda1=:false other parameters... -v
```

If you're using the command line tools on a Windows system, you must put quotation marks around the block device mapping value.

```
PROMPT> ec2-run-instances ami-1a2b3c4d -b "xvda=:false" other parameters... -v
```

By running the command in verbose mode, you can see the underlying request and response, and confirm that the `deleteOnTermination` value is set to `false`, as shown here.

```
...
<blockDeviceMapping>
  <item>
    <deviceName>/dev/sda1</deviceName>
    <ebs>
      <deleteOnTermination>false</deleteOnTermination>
    </ebs>
  </item>
</blockDeviceMapping>
...
```

For more information, see [ec2-run-instances](#).

Instance Best Practices

The instance is your basic computation building block. After you launch an instance, it 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 computer.

By default, you can run up to 20 instances. You can run as many or as few of these instances as you need at any time. If you need more than 20 instances, please complete the [Amazon EC2 Instance Request Form](#).

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, see the [Amazon CloudWatch product page](#).

- Keep your instances secure.

There are a number of ways you can keep your instances secure.

- Restrict access by only allowing trusted hosts or networks to access ports on your instance. For example, you can restrict SSH access by restricting incoming traffic on port 22. For more information about authorizing network access, see [Authorizing Network Access to Your Instances \(p. 293\)](#).
- Disable password-based logins for instances launched from your AMI. Passwords can be found or cracked, and are a security risk. For more information, see [Disable Password-Based Logins for Root \(p. 59\)](#). For more information about sharing AMIs safely, see [Sharing AMIs Safely \(p. 58\)](#).
- Review the rules in your security groups regularly, and ensure that you apply the principle of *least privilege*—only open up permissions that you require. You can also create different security groups to deal with instances that have different security requirements. Consider creating a bastion security group that allows external logins and keep the remainder of your instances in a group that does not allow external logins. For more information about security groups, see [Amazon EC2 Security Groups \(p. 415\)](#).
- Consider using AWS Identity and Access Management (IAM) to control access to your Amazon EC2 resources, including your instances. You can use IAM to create users and groups under your AWS account, assign security credentials to each, and control access to resources and services. For more information, see [Controlling Access to Amazon EC2 Resources \(p. 421\)](#).
- Stopping vs. terminating an instance

Stopping, Starting, and Terminating Instances

Stopped instances

When an instance is stopped, the instance will first perform a normal shutdown, 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.

You are not 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 after the instance is started. All of the associated Amazon EBS usage of your instance, including root device usage, are billed using the typical Amazon EBS prices.

When an instance is in a stopped state, you can attach or detach Amazon EBS volumes. You can use the `CreateImage` action to create an AMI from the instance. You can change the kernel, RAM disk, and instance type with the `ModifyInstanceAttribute` action. You can restart the instance with the `StartInstances` action, and terminate the instance with the `TerminateInstances` action.

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 using the instance limit request form.

Terminated instances

When an instance is terminated, the instance will first perform a normal shutdown, then the attached Amazon EBS volumes will be deleted if the volume's `deleteOnTermination` attribute is set to `true`. The instance itself will also be deleted, and the instance cannot be started again at a later date.

For increased safety, you may wish to disable termination using the `TerminateInstances` action. If you do so, 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 with Amazon EBS volume root devices default to `stop`, and instances with instance-store root devices are always terminated as the result of an instance shutdown.

Spot Instances

If you have flexibility on when your application will run, you can bid on unused Amazon EC2 compute capacity, called Spot Instances, and lower your costs significantly. Set by Amazon EC2, the Spot price for these instances fluctuates periodically depending on the supply of and demand for Spot Instance capacity.

To use Spot Instances, you place a Spot Instance request (your bid) specifying the maximum price you are willing to pay per hour per instance. If the maximum price of your bid is greater than the current Spot price, your request is fulfilled and your instances run until you terminate them or the Spot price increases above your maximum price. Your instance can also be terminated when your bid price equals the market price, even when there is no increase in the market price. This can happen when demand for capacity rises, or when supply fluctuates.

You will often pay less per hour than your maximum bid price. The Spot price is adjusted periodically as requests come in and the available supply of instances changes. Everyone pays that same Spot price for that period regardless of whether their maximum bid price was higher, and you will never pay more than your hourly maximum bid price.

Quick Look: Getting Started with Spot Instances Video

The following video shows how to launch your first Spot Instance using the AWS Management Console. This video includes instructions on placing a bid, determining when the instance is fulfilled, and canceling the instance. [Getting Started with Spot Instances](#)

Checklist for Getting Started with Spot Instances

If you want to get started working with Spot Instances, here are the resources you need to get going.

- [Getting Started with Spot Instances \(p. 133\)](#)
 - [Viewing Spot Price History \(p. 136\)](#)
 - [Creating a Spot Instance Request \(p. 139\)](#)
 - [Finding Running Spot Instances \(p. 143\)](#)
 - [Canceling Spot Instance Requests \(p. 147\)](#)
- [Fundamentals of Spot Instances \(p. 150\)](#)
 - [Placing Spot Requests \(p. 150\)](#)
 - [Tagging Spot Instance Requests \(p. 159\)](#)
 - [Protecting Your Spot Instance Data Against Interruptions \(p. 160\)](#)
- [Walkthroughs: Using Spot Instances with AWS Services \(p. 161\)](#)
 - [Managing Spot Instances with Auto Scaling \(p. 161\)](#)
 - [Using CloudFormation Templates to Launch Spot Instances \(p. 176\)](#)
 - [Launching Amazon Elastic MapReduce Job Flows with Spot Instances \(p. 177\)](#)
 - [Launching Spot Instances in Amazon Virtual Private Cloud \(p. 178\)](#)
- [Advanced Tasks \(p. 180\)](#)
 - [Subscribe to Your Spot Instance Data Feed \(p. 180\)](#)
 - [Programming Spot with AWS Java SDK \(p. 183\)](#)
 - [Starting Clusters on Spot Instances \(p. 210\)](#)

What's New in Spot Instances

Here's a quick look at what's new in Spot Instances:

- [Customizing Your Spot Requests \(p. 152\)](#)
- [Tracking Spot Requests with Bid Status Codes \(p. 153\)](#)

Getting Started with Spot Instances

In this section, we will step you through how to get started using Spot Instances. First, we'll take you through what you need to know before you begin and the prerequisites you need to get started.

We will walk you through the following activities:

- [Viewing Spot Price History \(p. 136\)](#)
- [Creating a Spot Instance Request \(p. 139\)](#)
- [Finding Running Spot Instances \(p. 143\)](#)
- [Canceling Spot Instance Requests \(p. 147\)](#)

Before You Begin

If you are new to Spot Instances, take a look at [Prerequisites for Using Spot Instances \(p. 133\)](#) to make sure you can take full advantage of the benefits of this Amazon EC2 product. If you have been using Amazon EC2 and you're ready to proceed, click one of the steps in the preceding list to get going.

Before requesting a Spot Instance, consider configuring your Amazon Machine Image (AMI) so that your application does the following:

- Automatically performs the tasks you want at start-up because the instance will start asynchronously without notification. For example, if you're using batch processes, you could set up your AMI to pull jobs from an Amazon Simple Queue Service (Amazon SQS) queue.
- Stores important data regularly in a place that won't be affected by instance termination. For example, you could store your data using Amazon Simple Storage Service (Amazon S3), Amazon SimpleDB, or Amazon Elastic Block Store (Amazon EBS).

Important

Although Spot Instances can use Amazon EBS-backed AMIs, be aware that Spot Instances do not support the Stop/Start feature. In other words, you can't stop and start Spot Instances launched from an AMI that has an Amazon EBS root device. For information about Amazon EBS, see [Amazon Elastic Block Store \(Amazon EBS\)](#).

- Handles termination gracefully.

For information about creating AMIs, see [Creating Your Own AMIs \(p. 30\)](#).

Prerequisites for Using Spot Instances

To work with Amazon EC2 Spot Instances, we assume you have read and completed the instructions described in [Getting Started with Amazon EC2 Linux Instances \(p. 8\)](#), which provides information on creating your Amazon EC2 account and credentials.

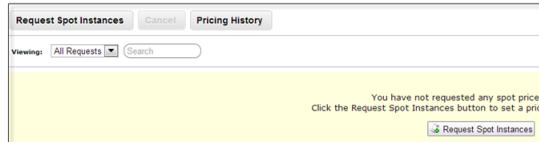
In addition, whichever you choose to use—the AWS Management Console, the Amazon EC2 command line interface (CLI), or the Amazon EC2 application programming interface (API)—Amazon EC2 provides tools for Spot Instances that you can use to assess Spot price history, submit Spot Instance requests

(also called *bids*), and manage your Spot requests and instances. You can also use, develop, and manage your applications using the AWS SDKs. For more information, see [Tools for Amazon Web Services](#).

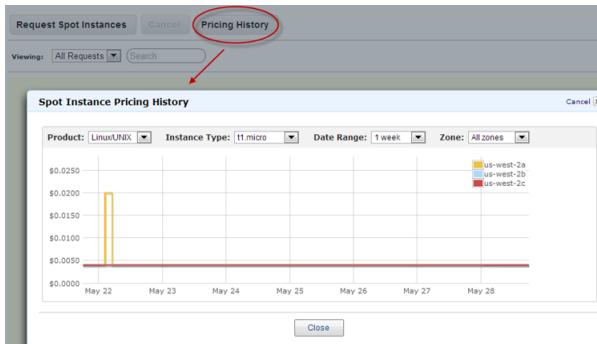
AWS Management Console

In the AWS Management Console, the EC2 console has tools specifically designed for Spot Instance request tasks. The EC2 console also has general tools that you can use to manage the instances launched when your requests are fulfilled. The following list identifies the tools you can use in the EC2 console:

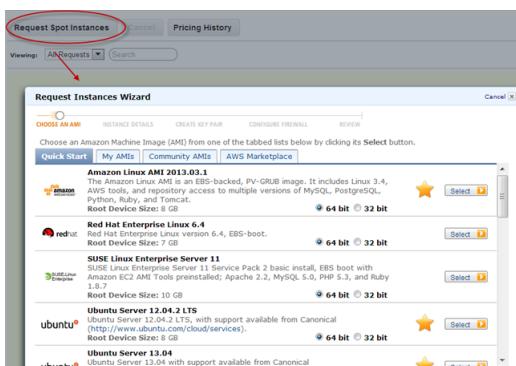
- The **Spot Requests** page is the main way you interact with your Spot Instance requests.



- Spot Instance Pricing History** gives you an insight in the pricing patterns for specific Spot Instance types in Availability Zones over a defined period.



- Use the **Request Spot Instances** page to submit a Spot Instance request and specify the details of the instance to be launched when your request succeeds.



- Use the **Instances** page to manage the instances launched when your Spot request succeeds.

Name	Instance	AMI ID	Type	State	Lifecycle
empty	i-dcaacde9	ami-0358ce33	t1.micro	running	spot

Amazon EC2 Command Line Interface (CLI) Tools

You can use Amazon EC2 CLI tools specifically designed for managing Spot requests. To manage the instances launched when your Spot request is fulfilled, use the same commands that you use for On-Demand EC2 instances.

Note

If you use the Amazon EC2 command line interface (CLI) tools, we assume that you have read and completed the instructions described in the [Setting Up the Amazon EC2 Command Line Interface Tools on Linux/UNIX \(p. 560\)](#). It walks you through setting up your environment for use with the CLI tools.

The following table lists the commands you use for Spot request tasks.

Task	CLI
Viewing Spot Price History (p. 136).	<code>ec2-describe-spot-price-history</code>
Finding Running Spot Instances (p. 143).	<code>ec2-describe-spot-instance-requests</code>
Creating a Spot Instance Request (p. 139).	<code>ec2-request-spot-instances</code>
Subscribe to Your Spot Instance Data Feed (p. 181).	<code>ec2-create-spot-datafeed-subscription</code>
Data Feed Filename and Format (p. 180).	<code>ec2-describe-spot-datafeed-subscription</code>
Delete a Spot Instance Data Feed (p. 182).	<code>ec2-delete-spot-datafeed-subscription</code>
Canceling Spot Instance Requests (p. 147).	<code>ec2-cancel-spot-instance-requests</code>

For information about CLI commands, go to the [Amazon Elastic Compute Cloud Command Line Reference](#).

API

You can use Amazon EC2 API tools specifically designed to manage your Spot requests. To manage the instances launched when your Spot request is fulfilled, use the same API actions that you use for On-Demand EC2 instances.

The following table lists the API actions you use for Spot request tasks.

Task	API
Viewing Spot Price History (p. 136).	<code>DescribeSpotPriceHistory</code>
Finding Running Spot Instances (p. 143).	<code>DescribeSpotInstanceRequests</code>
Creating a Spot Instance Request (p. 139).	<code>RequestSpotInstances</code>
Subscribe to Your Spot Instance Data Feed (p. 181).	<code>CreateSpotDatafeedSubscription</code>
Data Feed Filename and Format (p. 180).	<code>DescribeSpotDatafeedSubscription</code>
Delete a Spot Instance Data Feed (p. 182).	<code>DeleteSpotDatafeedSubscription</code>
Canceling Spot Instance Requests (p. 147).	<code>CancelSpotInstanceRequests</code>

For information about API actions, go to the [Amazon Elastic Compute Cloud API Reference](#).

AWS Java SDK

Java developers can go to the [AWS SDK for Java](#) to consult the Java tutorials on Spot Instances:

- [Tutorial: Amazon EC2 Spot Instances \(p. 184\)](#)
- [Tutorial: Advanced Amazon EC2 Spot Request Management \(p. 194\)](#)

Viewing Spot Price History

The Spot price represents the price above which you have to bid to guarantee that a single Spot request is fulfilled. When your bid is above the Spot price, your Spot Instance is launched, and if the Spot price rises above your bid price, your Spot Instance is terminated. You might choose to bid above the current Spot price so that your Spot request is fulfilled quickly. However, before specifying a price with which you want to bid for your Spot Instance, we recommend that you view the Spot price history. You can view the Spot price history for the last 90 days for any *pool* of Spot Instances sharing the same instance type, operating system, and Availability Zone.

For example, let's say you want to bid on a Linux/UNIX t1.micro instance to be launched in the us-east-1 region. To view past prices in this Spot pool, specify these values using the **Spot Price History** page of the AWS Management Console, the `DescribeSpotPriceHistory` API action, or `ec2-describe-spot-price-history` CLI command. If you need to launch your Spot Instance in a specific Availability Zone, you can specify that Availability Zone when retrieving the Spot price history.

After you review the Spot price history, you might choose to bid at a price that would have given you 75 percent Spot Instance uptime in the past. Or, you might choose to bid two times the current Spot price because doing so would have given you 99 percent uptime in the past. However you frame your bid, keep in mind that past performance of Spot prices is not a guarantee of future results. Spot prices vary based on real-time supply and demand conditions, and the conditions that generated certain Spot prices or pricing patterns in the past may not repeat themselves in the future.

Note

Make sure you have set up the prerequisites for working with Amazon EC2. If you haven't, see [Prerequisites for Using Spot Instances \(p. 133\)](#).

If you are using an API version earlier than 2011-05-15, the `DescribeSpotPriceHistory` action or the `ec2-describe-spot-price-history` command will return the lowest price across the region for the given time period and the prices will be returned in chronological order.

AWS Management Console

To view Spot Price history

1. From the [Amazon EC2 console](#), click **Spot Requests** in the navigation pane.

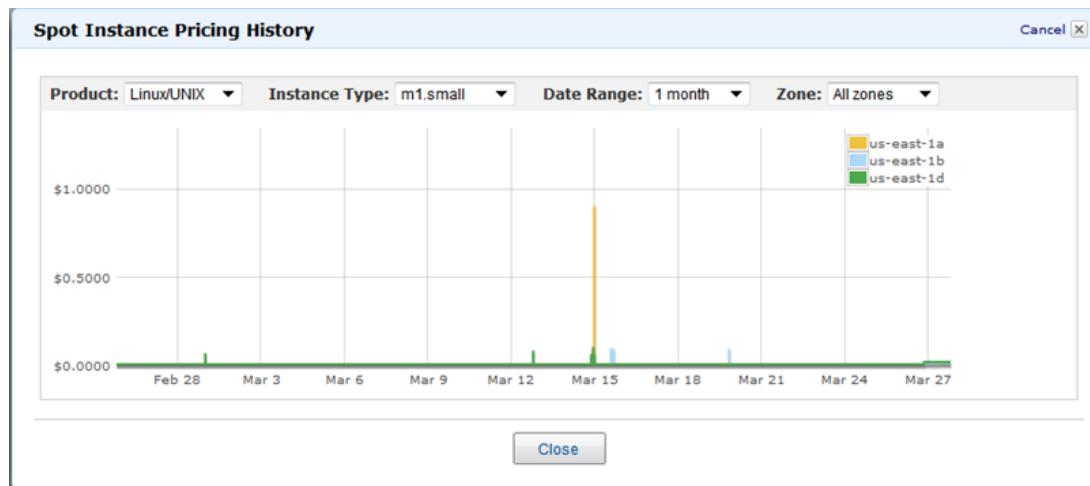
The **Spot Requests** pane opens on the right. It will list your Spot requests if you have any.

2. At the top of the pane, click the **Pricing History** button.

The console displays the **Spot Instance Pricing History** page.

3. If you want to view the Spot price history for specific Availability Zones, click the **Zone** drop-down list and select an Availability Zone.

The **Spot Instance Pricing History** page displays the Spot Instance pricing history for all zones or the zone you selected.



4. Using the price history as a guide, select a price that you think would likely keep your instances running for the period of time you need.

Amazon EC2 Command Line Interface (CLI) Tools

To view Spot price history

1. Enter the following command:

```
PROMPT> ec2-describe-spot-price-history -H --instance-type m1.xlarge
```

Amazon EC2 returns output similar to the following:

```
SPOTINSTANCERPRICE 0.384000 2011-05-25T11:37:48-0800 m1.xlarge
Windows us-east-1b
SPOTINSTANCERPRICE 0.384000 2011-05-25T11:37:48-0800 m1.xlarge
Windows us-east-1d
...
SPOTINSTANCERPRICE 0.242000 2011-04-18T14:39:14-0800 m1.xlarge SUSE
Linux us-east-1d
SPOTINSTANCERPRICE 0.242000 2011-04-18T14:39:14-0800 m1.xlarge SUSE
Linux us-east-1a
```

In this example, the price for the `m1.xlarge` instance type ranges between \$0.242 and \$0.384.

2. Using the price history as a guide, select a price that you think would likely keep your instances running for the period of time that you need.

Tip

You can filter the Spot history data so it includes only instance types or dates of interest to you. For more information about how to filter the results, go to [ec2-describe-spot-price-history](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

API

To view Spot price history

1. Construct the following Query request.

```
https://ec2.amazonaws.com/  
?Action=DescribeSpotPriceHistory  
&InstanceType=instance_type  
&...auth parameters...
```

Following is an example response.

```
<DescribeSpotPriceHistoryResponse xmlns="http://ec2.amazonaws.com/doc/2013-  
02-01/">  
    <spotPriceHistorySet>  
        <item>  
            <instanceType>m1.small</instanceType>  
            <productDescription>Linux/UNIX</productDescription>  
            <spotPrice>.28</spotPrice>  
            <timestamp>2009-12-01T11:51:50.000Z</timestamp>  
            <availabilityZone>&region_iad;&az1;</availabilityZone>  
        </item>  
        <item>  
            <instanceType>m1.small</instanceType>  
            <productDescription>Linux/UNIX</productDescription>  
            <spotPrice>.28</spotPrice>  
            <timestamp>2009-12-01T11:51:50.000Z</timestamp>  
            <availabilityZone>&region_iad;&az1;</availabilityZone>  
        </item>  
        <item>  
            <instanceType>m1.small</instanceType>  
            <productDescription>Linux/UNIX</productDescription>  
            <spotPrice>.31</spotPrice>  
            <timestamp>2009-12-01T11:51:50.000Z</timestamp>  
            <availabilityZone>&region_iad;&az2;</availabilityZone>  
        </item>  
        <item>  
            <instanceType>m1.small</instanceType>  
            <productDescription>Linux/UNIX</productDescription>  
            <spotPrice>.30</spotPrice>  
            <timestamp>2009-12-01T11:51:50.000Z</timestamp>  
            <availabilityZone>&region_iad;&az2;</availabilityZone>  
        </item>  
        <item>  
            <instanceType>m1.small</instanceType>  
            <productDescription>Linux/UNIX</productDescription>  
            <spotPrice>.25</spotPrice>  
            <timestamp>2009-12-01T11:51:50.000Z</timestamp>  
            <availabilityZone>&region_iad;&az3;</availabilityZone>  
        </item>  
        <item>  
            <instanceType>m1.small</instanceType>  
            <productDescription>Linux/UNIX</productDescription>  
            <spotPrice>.28</spotPrice>  
            <timestamp>2009-12-01T11:51:50.000Z</timestamp>  
            <availabilityZone>&region_iad;&az3;</availabilityZone>  
        </item>  
    </spotPriceHistorySet>  
</DescribeSpotPriceHistoryResponse>
```

```
</item>
<item>
  <instanceType>m1.small</instanceType>
  <productDescription>Linux/UNIX</productDescription>
  <spotPrice>.35</spotPrice>
  <tstamp>2009-12-01T11:51:50.000Z</tstamp>
  <availabilityZone>&region_iad;&az3;</availabilityZone>
</item>
</spotPriceHistorySet>
<nextToken/>
</DescribeSpotPriceHistoryResponse>
```

2. Using the price history as a guide, select a price that you think would likely keep your instances running for the period of time you need.

Tip

You can filter the Spot history data so it includes only instance types or dates of interest to you. For more information about how to filter the results, go to [DescribeSpotPriceHistory](#) in the *Amazon Elastic Compute Cloud API Reference*.

What do you want to do next?

- [Creating a Spot Instance Request \(p. 139\)](#)
- [Finding Running Spot Instances \(p. 143\)](#)
- [Launching Spot Instances in Amazon Virtual Private Cloud \(p. 178\)](#)

Creating a Spot Instance Request

After deciding on your bid price, you are ready to request a Spot Instance. Using the AWS Management Console, the API, or the CLI, you can specify how many Spot Instances you'd like, what instance type to use, and the bid price you are willing to pay.

If you request multiple Spot Instances at once, Amazon EC2 creates separate Spot Instance Requests with unique IDs (e.g., sir-1a2b3c4d) so that you can track the status of each request separately. For information about tracking Spot requests, see [Tracking Spot Requests with Bid Status Codes \(p. 153\)](#).

In this section, we discuss how to create requests for Amazon EC2 Spot Instances using the AWS Management Console, the API, or the CLI. For more information about requesting Spot Instances, see the following:

- [Spot Instance Limits \(p. 151\)](#)
- [Types of Spot Instance Requests \(p. 152\)](#)
- [Launching Spot Instances in Launch Groups and Availability Zones \(p. 153\)](#)
- [Spot Instance Request States \(p. 152\)](#)
- [Launching Spot Instances in Amazon Virtual Private Cloud \(p. 178\)](#)

Note

Make sure you have set up the prerequisites for working with Amazon EC2. If you haven't, see [Prerequisites for Using Spot Instances \(p. 133\)](#).

AWS Management Console

To create a Spot Instance request

1. From the [Amazon EC2 console](#), click **Spot Requests** in the navigation pane.
2. Click the **Request Spot Instances** button.

The **Request Instances Wizard** starts.

3. In the **Request Instances Wizard**, choose an AMI, and click **Select**.
4. On the **INSTANCE DETAILS** page, specify the number and type of instances you want. Then in the **Request Spot Instances** section, choose and specify the options you want to configure your Spot Instance settings, and click **Continue**.

The screenshot shows the 'Request Instances Wizard' interface. The 'INSTANCE DETAILS' tab is active. The 'Request Spot Instances' section is selected. It includes fields for 'Current Price' (\$0.004 - \$0.006), 'Max Price' (\$0.045), 'Request Valid From' (any time), 'Request Valid Until' (any time), 'Launch Into' (EC2-Classic selected), 'Availability Zone' (No Preference), and 'Availability Zone Group'. At the bottom are 'Back' and 'Continue' buttons.

Note

Amazon EC2 supports the following platforms: EC2-Classic and EC2-VPC. Your AWS account is capable of launching instances either into both platforms or only into EC2-VPC, on a region-by-region basis. If your account is capable of launching instances into both platforms, you will see the **EC2-Classic** and **EC2-VPC** tabs. If your account can launch instances only into EC2-VPC, you will not see any tab. Instead, you will see a **Subnet** drop-down box that lets you select where to launch your Spot Instance. For information, see [Supported Platforms \(p. 120\)](#).

Option	Description
Current Price	Displays the price for the Availability Zone you specified, or the lowest price across all Availability Zones if you do not specify an Availability Zone.
Max Price	Specifies the maximum price you are willing to pay per instance hour.

Option	Description
Request Valid From and Request Valid Until	Defines the validity period of a Spot request. The validity period, beginning with Request Valid From through Request Valid Until , specifies the length of time that your request will remain valid. By default, a Spot request will be considered for fulfillment from the time it is created until it is either fulfilled or canceled by you. However, you can constrain the validity period using these options. Note The end time you specify doesn't apply to the Spot Instances launched by this request. This end time only applies to the Spot Instance request.
Persistent Request?	Determines whether your request is one-time or persistent. By default, it is one-time. A one-time request can only be fulfilled once. A persistent request is considered for fulfillment whenever there is no Spot Instance running for the same request.
Launch Group	Groups a set of requests together in the same launch group. All requests in a launch group have their instances started and terminated together. For more information about using Launch Group in your Spot request, see Launching Spot Instances in Launch Groups and Availability Zones (p. 153) .
Availability Zone Group	Groups a set of requests together in the same Availability Zone. All requests that share an Availability Zone group will launch Spot Instances in the same Availability Zone. For more information about using Availability Zone Group in your Spot request, see Launching Spot Instances in Launch Groups and Availability Zones (p. 153) .

5. Continue with the wizard as you normally would when launching instances.

If your request specified multiple instances, Amazon EC2 creates a separate Spot Instance request for each instance. The requests are displayed on the **Spot Requests** page.

Amazon EC2 Command Line Interface (CLI) Tools

To create a Spot price request

- Enter the following command:

```
PROMPT> ec2-request-spot-instances
--price price
--user-data data
```

```
--instance-count count
--type one-time / persistent
[--valid-from timestamp]
[--valid-until timestamp]
[--launch-group launchgroup]
(run-instances-arguments)
```

For example:

```
PROMPT> ec2-request-spot-instances --price 0.32 ami-1234abcd --key MyKeypair
--group websrv --instance-type m1.small --instance-count 10 --type one-time
```

Amazon EC2 returns output similar to the following:

```
SPOTINSTANCEREQUEST      sir-09fb0a04      0.32      one-time      Linux/UNIX
    open      2010-04-06T10:03:09+0200      ami-1234abc      m1.small
MyKeypair      websrv      monitoring-disabled
...
```

If your request specified multiple instances, Amazon EC2 creates a separate Spot Instance request for each instance; each instance has a different ID (e.g., sir-09fb0a04).

API

To create a Spot price request

- Construct a Query request similar to the following.

```
https://ec2.amazonaws.com/
?Action=RequestSpotInstances
&SpotPrice.1=0.50
&InstanceCount.1=10
&Type.1=one-time
&AvailabilityZoneGroup.1=MyAzGroup
&LaunchSpecification.ImageId.1=ami-43a4412a
&LaunchSpecification.KeyName.1=MyKeypair
&LaunchSpecification.GroupSet.1=websrv
&LaunchSpecification.InstanceType.1=m1.small
&...auth parameters...
```

Following is an example response.

```
<RequestSpotInstancesResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
  <spotInstanceRequestSet>
    <item>
      <spotInstanceRequestId>sir-876aff12</spotInstanceRequestId>
      <spotPrice>0.32</spotPrice>
      <type>one-time</type>
      <state>open</state>
      <fault/>
      <validFrom/>
      <validUntil/>
```

```
<launchGroup/>
<availabilityZoneGroup>MyAzGroup</availabilityZoneGroup>
<launchSpecification>
    <imageId>ami-43a4412a</imageId>
    <keyName>MyKeypair</keyName>
    <groupSet>
        <item>
            <groupId>websrv</groupId>
        </item>
    </groupSet>
    <instanceType>m1.small</instanceType>
    ...
</launchSpecification>
<instanceId/>
<createTime>2009-10-19T00:00:00+0000</createTime>
<productDescription>Linux/UNIX</productDescription>
<tagSet/>
</item>
...
</spotInstanceRequestSet>
</RequestSpotInstancesResponse>
```

If your request specified multiple instances, Amazon EC2 creates a separate Spot Instance request for each instance; each instance has a different ID (e.g., sir-876aff12).

What do you want to do next?

- [Finding Running Spot Instances \(p. 143\)](#)
- [Canceling Spot Instance Requests \(p. 147\)](#)
- [Planning for Interruptions \(p. 160\)](#)
- [Launching Spot Instances in Amazon Virtual Private Cloud \(p. 178\)](#)
- [Tagging Spot Instance Requests \(p. 159\)](#)
- [Subscribe to Your Spot Instance Data Feed \(p. 180\)](#)

Finding Running Spot Instances

Your Spot request will launch a Spot Instance when the Spot price is below your bid price. A Spot Instance will run until either its maximum bid price is no longer higher than the Spot price or you terminate the Spot Instance yourself. (If your bid price is exactly equal to the Spot price, there is a chance your Spot Instance will remain running.) When your Spot request is fulfilled and has running instances, your Spot request will be *active* (as opposed to *open*, *closed*, or *canceled*). In this section, we describe how to find running Spot Instances associated with your active Spot requests.

Note

Make sure you have set up the prerequisites for working with Amazon EC2. If you haven't, go to [Prerequisites for Using Spot Instances \(p. 133\)](#).

AWS Management Console

To find running Spot Instances

1. From the [Amazon EC2 console](#), click **Instances** in the navigation pane.

The console displays a list of running instances.

Launch Instance						
Viewing:		Actions				
	Name	Instance	AMI ID	Type	State	Lifecycle
	empty	i-dcaacde9	ami-0358ce33	t1.micro	running	spot

2. To identify which of the instances were launched as a result of Spot requests, match the Instance IDs in the **Instances** page—where the **Lifecycle** column lists the instance as *spot*—with Instance IDs in the **Spot Requests** page. (You might need to turn on the display of the column by clicking **Show/Hide** in the top right corner.)

Request ID	Max Price	AMI ID	Instance	Type	State	Status	Persistence	Launch Group	Zone Group
sir-1ba42234	\$0.008	ami-3275ee5b	i-496eaad5	t1.micro	active	fulfilled	persistent		
sir-3891634	\$0.008	ami-3275ee5b	i-48cd4829	t1.micro	active	fulfilled	persistent		
sir-e7e3d34	\$0.008	ami-3275ee5b	i-58c6d79	t1.micro	active	fulfilled	persistent		

Name	Instance	AMI ID	Type	State	Lifecycle
empty	i-58c05839	ami-3275ee5b	t1.micro	running	spot
empty	i-63279309	ami-3275ee5b	t1.micro	running	spot
empty	i-d22caeb1	ami-3275ee5b	t1.micro	running	spot
empty	i-65707202	ami-3275ee5b	t1.micro	running	spot
empty	i-47e65a9	ami-3275ee5b	t1.micro	running	spot
empty	i-48cd4829	ami-3275ee5b	t1.micro	running	spot
empty	i-c496a3a1	ami-3275ee5b	m1.small	running	normal
empty	i-c496a3a5	ami-3275ee5b	m1.small	running	normal
empty	i-c496a3a7	ami-3275ee5b	m1.small	running	normal

Amazon EC2 Command Line Interface (CLI) Tools

To find running Spot Instances

1. Use `ec2-describe-spot-instance-requests`. If your Spot Instance request has been fulfilled (an instance has been launched), the instance ID appears in the response.

```
PROMPT> ec2-describe-spot-instance-requests
SPOTINSTANCEREQUEST      sir-e1471206      0.09      one-time      Linux/UNIX
      active   2010-09-13T16:50:44-0800
i-992cf7dd      ami-813968c4      m1.small      MyKey      default
      monitoring-disabled
```

2. Alternatively, you can use `ec2-describe-instances` with the following filter: `--filter instance-lifecycle=spot`. If you're using the command line interface tools on a Windows system, you might need to use quotation marks (`--filter "instance-lifecycle=spot"`). For more information about filters, see [Listing and Filtering Your Resources \(p. 535\)](#).

```
PROMPT> ec2-describe-instances --filter instance-lifecycle=spot
```

Amazon EC2 returns output similar to the following:

```
RESERVATION      r-b58651f1      111122223333      default
INSTANCE        i-992cf7dd      ami-813968c4      ec2-184-72-8-111.us-west-
1.compute.amazonaws.com      ip-10-166-105-139.us-west-1.compute.internal
      running      MyKey      0      m1.small      2010-09-13T23:54:40+0000
      us-west-1a      aki-a13667e4      ari-a33667e6
```

monitoring-disabled 184.72.8.111 10.166.105.139 spot sir-e1471206 paravirtual	ebs
-----------------------------------------------------------------------------------------	-----

API

To find running Spot Instances

1. Construct a `DescribeInstances` Query request and use a filter to look for instances where `instance-lifecycle=spot`. For more information about filters, see [Listing and Filtering Your Resources \(p. 535\)](#).

```
https://ec2.amazonaws.com/  
?Action=DescribeInstances  
&Filter.1.Name=instance-lifecycle  
&Filter.1.Value.1=spot  
&...auth parameters...
```

Following is an example response. It includes an `instanceLifecycle` element with `spot` as the value.

```
<DescribeInstancesResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">  
...  
<instancesSet>  
<item>  
  <instanceId>i-992cf7dd</instanceId>  
  <imageId>ami-813968c4</imageId>  
  <instanceState>  
    <code>16</code>  
    <name>running</name>  
  </instanceState>  
  <privateDnsName>ip-10-166-105-139.&region_sfo;.compute.internal</privateDnsName>  
  <dnsName>ec2-184-72-8-111.&region_sfo;.compute.&api-domain;</dnsName>  
  <reason/>  
  <keyName>MyKey</keyName>  
  <amiLaunchIndex>0</amiLaunchIndex>  
  <productCodes/>  
  <instanceType>m1.small</instanceType>  
  <launchTime>2010-09-13T23:54:40.000Z</launchTime>  
  <placement>  
    <availabilityZone>&region_sfo;&az1;</availabilityZone>  
    <groupName/>  
  </placement>  
  <kernelId>aki-a13667e4</kernelId>  
  <ramdiskId>ari-a33667e6</ramdiskId>  
  <monitoring>  
    <state>disabled</state>  
  </monitoring>  
  <privateIpAddress>10.166.105.139</privateIpAddress>  
  <ipAddress>184.72.8.111</ipAddress>  
  <architecture>i386</architecture>  
  <rootDeviceType>ebs</rootDeviceType>  
  <rootDeviceName>/dev/sda1</rootDeviceName>
```

```
<blockDeviceMapping>
  <item>
    <deviceName>/dev/sda1</deviceName>
    <ebs>
      <volumeId>vol-61088f0a</volumeId>
      <status>attached</status>
      <attachTime>2010-09-13T23:54:42.000Z</attachTime>
      <deleteOnTermination>true</deleteOnTermination>
    </ebs>
  </item>
</blockDeviceMapping>
<instanceLifecycle>spot</instanceLifecycle>
<spotInstanceRequestId>sir-e1471206</spotInstanceRequestId>
<virtualizationType>paravirtual</virtualizationType>
</item>
</instancesSet>
</DescribeInstancesResponse>
```

2. Alternatively, you can use `DescribeSpotInstanceRequests`. If your Spot Instance request has been fulfilled (an instance has been launched), the instance ID appears in the response. Following is an excerpt from a response.

```
...
<spotInstanceRequestSet>
  <item>
    <spotInstanceRequestId>sir-e1471206</spotInstanceRequestId>
    <spotPrice>0.09</spotPrice>
    <type>one-time</type>
    <state>active</state>
    <launchSpecification>
      <imageId>ami-813968c4</imageId>
      <keyName>MyKey</keyName>
      <groupSet>
        <item>
          <groupId>default</groupId>
        </item>
      </groupSet>
      <instanceType>m1.small</instanceType>
      <blockDeviceMapping/>
      <monitoring>
        <enabled>false</enabled>
      </monitoring>
    </launchSpecification>
    <instanceId>i-992cf7dd</instanceId>
    <createTime>2010-09-13T23:50:44.000Z</createTime>
    <productDescription>Linux/UNIX</productDescription>
    <launchedAvailabilityZone>&region_iad;&az3;</launchedAvailabilityZone>
  </item>
<spotInstanceRequestSet/>
...
```

What do you want to do next?

- [Canceling Spot Instance Requests \(p. 147\)](#)

- [Launching Spot Instances in Amazon Virtual Private Cloud \(p. 178\)](#)
- [Persisting Your Root EBS Partition \(p. 160\)](#)
- [Tagging Spot Instance Requests \(p. 159\)](#)
- [Subscribe to Your Spot Instance Data Feed \(p. 180\)](#)

Canceling Spot Instance Requests

If you no longer want your Spot request, you can cancel it. You can only cancel Spot Instance requests that are open or active. Your Spot request is *open* when your request has not yet been fulfilled and no instances have been launched. Your Spot request is *active* when your request has been fulfilled, and Spot Instances have launched as a result. If your Spot request is active and has an associated running Spot Instance, canceling the request does not automatically terminate the instance: You must terminate the running Spot Instance manually.

Note

Make sure you have set up the prerequisites for working with Amazon EC2. If you haven't, go to [Prerequisites for Using Spot Instances \(p. 133\)](#).

Your Spot request can be in other states in addition to open and active. For more information, see [Spot Instance Request States \(p. 152\)](#). You can then track the status of each of the requests separately. For information about tracking Spot requests, see [Tracking Spot Requests with Bid Status Codes \(p. 153\)](#).

A *persistent* Spot request will try to relaunch a Spot Instance whenever it is interrupted or terminated. If you want to end both the Spot request and its Spot Instance, make sure you cancel the Spot request before you terminate the Spot Instance to ensure that the request doesn't launch a new instance.

AWS Management Console

To cancel Spot Instance requests

1. From the [Amazon EC2 console](#), click **Spot Requests** in the navigation pane.

The console displays a list of Spot Instance requests.

	Request ID	Max Price	AMI ID	Instance	Type	State	Status
<input type="checkbox"/>	sir-280a7a34	\$0.050	ami-e565ba8c	i-8a2b30e6	m1.small	● active	fulfilled
<input type="checkbox"/>	sir-b0277635	\$0.050	ami-e565ba8c	i-a31bb2ce	m1.small	● active	fulfilled
<input type="checkbox"/>	sir-95335234	\$0.070	ami-e565ba8c	i-b6b021da	m1.small	● active	fulfilled

2. Select the Spot requests you want to cancel and click the **Cancel** button.
3. If you don't want the associated Spot Instances to continue running, you can cancel them. Select them in the **Instances** page, then right-click and select **Terminate**.

Name	Instance	AMI ID	Type	State	Lifecycle
<input checked="" type="checkbox"/> empty	i-08681864	ami-3275ea5b	t1.micro	● running	spot
<input type="checkbox"/> empty	i-6327930	Instance Management	cro	● running	spot
<input type="checkbox"/> empty	i-1d2ceff	Connect	cro	● running	spot
<input type="checkbox"/> empty	i-6870720	Get System Log	cro	● running	spot
<input type="checkbox"/> empty	i-c17e868	Create Image (EBS AMI)	cro	● running	spot
<input type="checkbox"/> empty	i-811f194	Add/Edit Tags	cro	● running	spot
<input type="checkbox"/> empty	i-ca86a3a	Change Security Groups	cro	● running	spot
<input type="checkbox"/> empty	i-ce86a3a	Change Source / Dest Check	cro	● running	spot
<input type="checkbox"/> empty	i-cc86a3a	Launch More Like This	mall	● running	normal
<input type="checkbox"/> empty	i-cc86a3a	Disassociate IP Address	mall	● running	normal
<input type="checkbox"/> empty	i-cc86a3a	Change Termination Protection	mall	● running	normal
<input type="checkbox"/> empty	i-cc86a3a	View/Change User Data	mall	● running	normal
<input type="checkbox"/> empty	i-cc86a3a	Change Instance Type			
<input type="checkbox"/> empty	i-cc86a3a	Change Shutdown Behavior			
<input type="checkbox"/> empty	i-cc86a3a	Attach Network Interface			
<input type="checkbox"/> empty	i-cc86a3a	Detach Network Interface			
<input type="checkbox"/> empty	i-cc86a3a	Manage Private IP Addresses			
		Instance Lifecycle			
		Terminate			
		Reboot			
		Stop			
		Start			

Amazon EC2 Command Line Interface (CLI) Tools

To cancel Spot Instance requests

1. Enter the following command to see your Spot Instance requests:

```
PROMPT> ec2-describe-spot-instance-requests
```

Amazon EC2 returns output similar to the following:

```
SPOTINSTANCEREQUEST    sir-09fb0a04    0.04    one-time    Linux/UNIX    closed
2010-04-06T10:03:09+0200    ami-b232d0db    m1.small    gsg-keypair    default
monitoring-disabled
```

2. To cancel the request, enter the following:

```
PROMPT> ec2-cancel-spot-instance-requests sir-09fb0a04
```

Amazon EC2 returns output similar to the following:

```
SPOTINSTANCEREQUEST    sir-09fb0a04    canceled
```

3. To terminate the instances associated with the Spot request you just canceled, enter the following:

```
PROMPT> ec2-terminate-instances i-48cd4829
```

Amazon EC2 returns output similar to the following:

```
INSTANCE    i-48cd4829    running    shutting-down
```

Tip

You can filter the list of Spot Instance requests to return only certain EC2 instance types. For more information about how to filter the results, go to [ec2-describe-spot-instance-requests](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

API

To cancel Spot Instance requests

1. Construct the following Query request to see your Spot Instance requests.

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

Following is an example response.

```
<DescribeSpotInstanceRequestsResponse xmlns="http://ec2.amazon  
aws.com/doc/2013-02-01/">  
  <spotInstanceRequestSet>  
    <item>  
      <spotInstanceId>sir-8675309a</spotInstanceId>  
      <spotPrice>0.32</spotPrice>  
      <type>one-time</type>  
      <state>open</state>  
      <fault/>  
      <validFrom/>  
      <validUntil/>  
      <launchGroup/>  
      <availabilityZoneGroup>MyAzGroup</availabilityZoneGroup>  
      <launchSpecification>  
        <imageId>i-43a4412a</imageId>  
        <keyName>MyKeypair</keyName>  
        <groupSet>websrv</groupSet>  
        <instanceType>m1.small</instanceType>  
      </launchSpecification>  
      <instanceId>i-123345678</instanceId>  
      <createTime>2009-10-19T00:00:00+0000</createTime>  
      <productDescription>Linux/UNIX</productDescription>  
      <launchedAvailabilityZone>&region_iad;&az3;</launchedAvailabilityZone>  
    </item>  
  </spotInstanceRequestSet>  
</DescribeSpotInstanceRequestsResponse>
```

2. Construct a Query request to cancel the Spot Instance requests.

```
https://ec2.amazonaws.com/  
?Action=CancelSpotInstanceRequests  
&SpotInstanceRequestId.1=sir-8675309a  
&...auth parameters...
```

Following is an example response.

```
<CancelSpotInstanceRequestsResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">  <spotInstanceRequestId>sir-8675309a</spotInstanceRequestId></CancelSpotInstanceRequestsResponse>
```

Tip

You can filter the list of Spot Instance requests to return only certain EC2 instance types. For more information about how to filter the results, go to [DescribeSpotInstanceRequests](#) in the *Amazon Elastic Compute Cloud API Reference*.

What do you want to do next?

- [Creating a Spot Instance Request \(p. 139\)](#)
- [Viewing Spot Price History \(p. 136\)](#)
- [Finding Running Spot Instances \(p. 143\)](#)
- [Launching Spot Instances in Amazon Virtual Private Cloud \(p. 178\)](#)

Fundamentals of Spot Instances

If your application can tolerate interruptions and can handle dynamically changing compute capacity and prices, Amazon EC2 Spot Instances can be a powerful tool for optimizing your application's costs and increasing its computational scale. Because Spot Instances are often offered at a large savings compared to regular On-Demand instances, you might be able to take advantage of Spot Instances to both accelerate your computational job and lower its overall cost.

This section presents the basics you need to know so you can create successful Spot requests, manage Spot Instances effectively, and plan for contingencies such as interruptions and changes in capacity.

- A Spot Instance request (also called a *bid*) defines the kind of Spot Instances you want and the maximum price you're willing to pay for them. How you specify your bid determines the fulfillment of the request and the successful launch of Spot Instances. For information, see [Placing Spot Requests \(p. 150\)](#).
- You can use tags to manage your Spot Instances. For information, see [Tagging Spot Instance Requests \(p. 159\)](#).
- The availability of Spot Instances varies significantly depending on Amazon EC2 capacity. When you use Spot Instances, you should expect and plan for interruptions and mitigate their effect on your applications. For information, see [Protecting Your Spot Instance Data Against Interruptions \(p. 160\)](#).

Placing Spot Requests

The Spot Instance request (or *bid*) is a core element in Amazon EC2 Spot Instances. Your specifications for a Spot request—price, Availability Zone, launch group, and schedule—affect the fulfillment of the request and the successful launch of Spot Instances.

This section discusses the bid fundamentals you need to know to help you manage your Spot requests.

- [Spot Instance Limits \(p. 151\)](#)

Discusses the number of Spot Instances per instance type that you can launch for each region.

- [Customizing Your Spot Requests \(p. 152\)](#)

Discusses the options you can specify to customize your Spot request.

- [Tracking Spot Requests with Bid Status Codes \(p. 153\)](#)

Discusses Spot bid status information and how you can use it to understand what is going on with your Spot requests.

Depending on how quickly you want your Spot request fulfilled and how long you want your Spot Instances to continue running, there are a number of factors you need to consider when you bid on Spot Instances. The requirements you include in your Spot request can affect the chances that it will be fulfilled. If you have a number of requirements that must be met, your request can take longer to fulfill.

The maximum Spot Instance price you're willing to pay—your *bid price*—may also affect how long your Spot Instances will run. If you want your instances to run as long as possible and have a low probability of interruption, you might consider submitting a higher bid price. If you are less concerned about Spot interruptions and simply want to take advantage of Spot cost savings, you should submit a lower bid price.

Your bid price is not necessarily the price that you pay. For example, if you bid \$0.500, and the Spot price is \$0.300 per hour for the period, you only pay \$0.300 per hour. If the Spot price drops, you pay less. If the Spot price increases, you pay the new price up to your bid price. If the Spot price rises above your bid price (or another capacity constraint is encountered), your Spot Instance is interrupted.

Note

Bidding high *does not guarantee* that your Spot Instance won't be interrupted. Also, do not assume that the Spot price can never remain high for long periods of time. If Spot supply is tight for an extended period of time, the Spot price can be elevated for a sustained period of time, pegged to the bid price of the highest bidders.

Quick Look: Bidding Strategies Video

The following video describes strategies to use when bidding for Spot Instances. [Deciding on Your Spot Bidding Strategies](#)

After you submit your bid for Spot Instances, use bid status codes to track your request. For information, see [Tracking Spot Requests with Bid Status Codes \(p. 153\)](#).

Spot Instance Limits

Generally, you are limited to running a total of 100 Spot Instances per region. Certain instance types have a different limit amount per region as shown in the following table:

Note

The instance types *cc2.8xlarge*, *cg1.4xlarge*, and *cr1.8xlarge* are not available in all regions.

Instance Type	Spot Limit
Cluster GPU Quadruple Extra Large (cg1.4xlarge)	10
High-Memory Double Extra Large (m2.2xlarge)	20
High-Memory Quadruple Extra Large (m2.4xlarge)	20
Cluster Compute Quadruple Extra Large (cc1.4xlarge)	20
Cluster Compute Eight Extra Large (cc2.8xlarge)	20
High I/O Quadruple Extra Large (hi1.4xlarge)	Not Offered
High Storage Eight Extra Large (hs1.8xlarge)	Not Offered
All Other Instance Types	100

If you need more Spot Instances, complete the [Amazon EC2 instance request form](#) with your use case and your instance increase will be considered. Limit increases are tied to the region specified in the Spot Instance request.

Customizing Your Spot Requests

You can use a number of options to customize your Spot requests so you get instances that address your needs. These options are discussed in the following topics:

- [Types of Spot Instance Requests \(p. 152\)](#)
- [Launching Spot Instances in Launch Groups and Availability Zones \(p. 153\)](#)

The options you choose can affect the success of your Spot requests as well as the lifespan of the Spot Instances that these requests launch.

Spot Instance Request States

A Spot Instance request can be in one of the following states:

- **Open**—The request is not fulfilled.
- **Active**—The request is currently active (fulfilled) and has an associated Spot Instance.
- **Failed**—The request failed because bad parameters were specified.
- **Closed**—The request either completed (a Spot Instance was launched and subsequently was interrupted or terminated), or was not fulfilled within the period specified.
- **Canceled**—The request is canceled because one of two events took place: You canceled the request, or the bid request went past its expiration date.

Depending on conditions, an instance might still be running even if the request is closed or canceled.

The **valid until** option that you specify when you submit a Spot Instance request applies only to the *request*; this deadline doesn't apply to Spot Instances that are *launched* by the request. This means that when your Spot Instance request is canceled, either because the request went beyond its expiration date or because you manually canceled it, the Spot Instances that were launched previously through the now-canceled request don't automatically get terminated. The Spot Instance service will terminate a running instance only when the Spot price equals or exceeds your price. However, you can always terminate your instances manually.

You can track the status of your Spot Instance requests as well as the status of the instances launched through bid status codes. For information about tracking Spot requests, see [Tracking Spot Requests with Bid Status Codes \(p. 153\)](#).

Types of Spot Instance Requests

You can make two types of Spot Instance requests—a one-time request or a persistent request. A one-time request remains active until one of the following conditions is met: All of the requested instances launch, the request expires, or you cancel the request. For example, if you create a one-time request for three instances, the request is considered complete after all three instances launch.

Persistent Spot Instance requests remain active until they expire or you cancel them, even if the requests were previously satisfied. For example, if you create a persistent Spot Instance request for one instance at \$0.300, Amazon EC2 launches or keeps your instance running if your maximum bid price is above \$0.300 and terminates your instance if your maximum bid price is below \$0.300.

With both one-time and persistent requests, instances continue to run until they no longer exceed the Spot price, you terminate them, or the instances terminate on their own. If the maximum price is exactly equal to the Spot price, an instance might or might not continue running (depending on available capacity).

Launching Spot Instances in Launch Groups and Availability Zones

You can opt to have your Spot Instances launch at the same time or in the same Availability Zone. To tell Amazon EC2 to launch your instances only if all the instances in the request can be fulfilled, specify a launch group in your Spot Instance request.

To specify a launch group using the Amazon EC2 console, you select **Launch Group** in the Request Instances Wizard. If you use the CLI or API tools, you specify the `launch_group` option when you use the `ec2-request-spot-instances` command or call the `RequestSpotInstances` action.

If you want all your instances to be launched together in a single Availability Zone, specify an Availability Zone group. Do this by selecting **Availability Zone Group** when using the Request Instances Wizard in the Amazon EC2 console, or by specifying the `availability-zone-group` option when you use the `ec2-request-spot-instances` CLI command or call the `RequestSpotInstances` API action.

Although the launch group and Availability Zone group specifications can be advantageous, use them only when needed. Avoiding these requirements increases the chances that your Spot Instance request can be fulfilled. Also, specifying a launch group increases the chance that your Spot Instances will be terminated: When one instance in the launch group is terminated, all of the instances in the group are terminated.

To identify a bid price for a Spot Instance in a specific Availability Zone, see [Viewing Spot Price History \(p. 136\)](#) for the price over time in the Availability Zone where you want to make your bid. If you use the Amazon EC2 console, go to the [Spot Instance Pricing History](#) page and select the Availability Zone from the **Zone** drop-down menu. You can also call `DescribeSpotPriceHistory` API action or use the `ec2-describe-spot-price-history` CLI command. You will see the price history for the specified Availability Zone with the most recent set of prices listed first. If you don't specify an Availability Zone, you will get the lowest prices across all Availability Zones for the time period, starting with the most recent set of prices.

Note

When you use the `DescribeSpotPriceHistory` action or the `ec2-describe-spot-price-history` command before the 2011-05-15 API version, you will get the lowest price across the region for the given time period and the prices will be returned in chronological order—that is, from the oldest to the most recent.

In addition, keep in mind when reviewing Spot prices that past performance is not a guarantee of future results. Spot prices vary based on real-time supply and demand conditions, and the conditions that generated certain Spot prices or pricing patterns in the past may not repeat themselves in the future.

For more information about Availability Zones, see [Region and Availability Zone Concepts \(p. 116\)](#).

Tracking Spot Requests with Bid Status Codes

Topics

- [The Life Cycle of a Spot Request \(p. 154\)](#)
- [Spot Bid Status Code Reference \(p. 157\)](#)

Bid statuses can help you track your Amazon Elastic Compute Cloud (EC2) Spot Instance requests, plan your use of Spot Instances, and bid strategically. Spot bid status information is composed of a status code, the update time, and a status message. Together, they help you determine the disposition of your Spot request. For example, a bid status can tell you the reason why your Spot request isn't fulfilled yet or list the constraints that are impeding the fulfillment of your Spot request.

Note

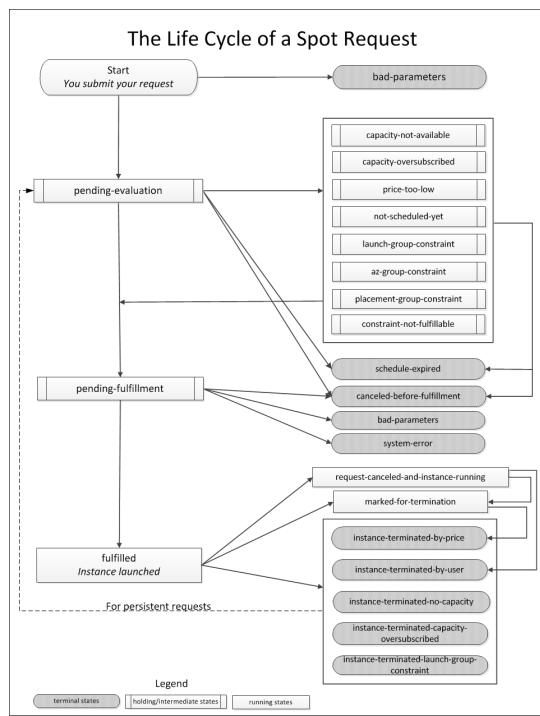
A *bid* is a *request* that is conditionally tied to a price. The terms are used interchangeably in this documentation.

You can use any of the Amazon EC2 Spot Instance tools to obtain status information about your Spot requests. Spot bid status information will be available for new Spot requests as well as bid requests that you already have open. However, if you're using either the Amazon EC2 CLI or API tools to obtain bid status information, you need to use version 2012-10-01 or later of the API tools. The following table summarizes how you can get bid status information using the Amazon EC2 tools.

Tool	Use	To get
Spot Requests page in the Amazon EC2 console	Status column	Status code
	Bottom pane, tooltip	Status message
Amazon EC2 CLI	ec2-request-spot-instances or ec2-describe-spot-instance-requests	Status code, status message, and update time
Amazon EC2 API	RequestSpotInstances or DescribeSpotInstanceRequests	Status code, status message, and update time

The Life Cycle of a Spot Request

This diagram shows you the paths your Spot request can follow throughout its life cycle. Following the diagram, we explain each step of the life cycle—from making a request to the terminal state.



When you request a Spot Instance, your request goes through an evaluation process. At each step of the process—also called the Spot request *life cycle*—specific events dictate successive request states. In the life cycle diagram, each step is depicted as a node, and each node's status code describes the status of your Spot request and Spot Instance.

- Pending evaluation** – As soon as you make your Spot Instance request, it goes into the pending-evaluation state.

Status Code	Request State	Instance State
pending-evaluation	open	(none)

The only reason your Spot bid request does not proceed to this pending-evaluation state is that at least one of the parameters is invalid (bad-parameters).

Status Code	Request State	Instance State
bad-parameters	closed	(none)

- Pending evaluation holding state** – Your Spot Instance request will report a status code in the *holding* state if one or more options you specified are valid but not met, or if we encounter a Spot capacity constraint. The options defined by your request affect the likelihood of it being fulfilled. For example, if you specified a max bid price below the current Spot price, or requested that your Spot Instances be launched together in one Availability Zone (AZ), your Spot request will stay in a *holding* state until the Spot price goes below your bid or the Availability Zone constraint is met.

The following table lists the status codes reported when your bid is in a *holding state* waiting for a constraint to be met.

Status Code	Request State	Instance State
capacity-not-available	open	(none)
capacity-oversubscribed	open	(none)
price-too-low	open	(none)
not-scheduled-yet	open	(none)
launch-group-constraint	open	(none)
az-group-constraint	open	(none)
placement-group-constraint	open	(none)
constraint-not-fulfillable	open	(none)

- Pending evaluation/fulfillment terminal state** – Before your request reaches the pending fulfillment phase, your Spot Instance request may end up in a *terminal* state if the valid dates you specified for the bid expired, you canceled the bid yourself, or a system error occurred.

Note

A *canceled* Spot request does not necessarily mean that the Spot Instance is terminated. In contrast, a *closed* Spot request means the request will no longer be considered and the Spot Instance is terminated.

The following table lists the status codes reported when your bid ends up in the *terminal* state after pending evaluation or pending fulfillment.

Status Code	Request State	Instance State
schedule-expired	closed	(none)
canceled-before-fulfillment*	canceled	(none)
bad-parameters	failed	(none)
system-error	closed	(none)

* This status results from a cancellation initiated by a user.

4. **Pending fulfillment** – When the constraints you specified (if any) are met, and your bid price is equal to or higher than the current Spot market price, your Spot request goes into the `pending-fulfillment` state.

Status Code	Request State	Instance State
<code>pending-fulfillment</code>	open	(none)

At this point, the Amazon EC2 Spot service is getting ready to provision the instances that you requested. If the process stops at this point, it would likely be due to cancellation by the user before a Spot Instance was launched, or an unexpected system error (see *Pending evaluation/fulfillment terminal state*).

5. **Fulfilled** – When all the specifications for your Spot Instance are met, your Spot request is fulfilled. The Amazon EC2 Spot service launches the Spot Instance, which may take a few minutes.

Status Code	Request State	Instance State
<code>fulfilled</code>	active	starting up → running

At this point, your instance launches, and your bid request is `active`. Your instance will continue to run as long as your bid price is at or above the Spot market price, EC2 has spare Spot capacity for your instance type, and you don't terminate the instance.

Note

If your bid price is equal to the market price, your request can stay `fulfilled`, but you run the risk of your Spot Instance getting terminated at any time that the market price goes up. In some cases, your Spot Instance could be evicted even if your bid equaled the Spot price because Spot Instances were oversubscribed at that price. In such cases, you will get the `instance-terminated-capacity-oversubscribed` status code. See *Fulfilled terminal state*.

6. **Fulfilled terminal state** – Your Spot request will report a status code in the *terminal* state if a change in Spot price or capacity requires the Spot service to terminate your Spot Instance. You will also get a terminal state status code if you cancel the Spot request or terminate the Spot Instance.

The following table lists the status codes reported when your instance is in the *terminal* state after the bid was fulfilled.

Status Code	Request State	Instance State
<code>request-canceled-and-instance-running</code>	<code>canceled[c]</code>	running

Status Code	Request State	Instance State
marked-for-termination	closed	running
instance-terminated-by-price	closed[a]	(Spot) terminated
instance-terminated-by-user (or spot-instance-terminated-by-user, which will be deprecated)	closed[b]	terminated[c]
instance-terminated-no-capacity	closed[a]	(Spot) terminated
instance-terminated-capacity-oversubscribed	closed[a]	(Spot) terminated
instance-terminated-launch-group-constraint	closed[a]	(Spot) terminated

[a] If the Spot bid request is persistent, the state of the request becomes *open*. See the next item, Persistent requests.

[b] If you terminate your instance first, there may be a delay before the Spot service realizes that your Spot Instance was terminated. For persistent requests, this means that for user-terminated instances it could take some time before your Spot request re-enters the pending-evaluation state.

[c] Actions are initiated by the user.

7. **Persistent requests** – When your Spot Instances are terminated (either by you or the Amazon EC2 Spot service), if you specified a *persistent* Spot request, then the request will be considered again. This request returns to the pending-evaluation state. Your Spot bid request will undergo the same evaluation process that it originally went through, and an entirely new Spot Instance will be launched for the persistent Spot request.

Spot Bid Status Code Reference

The following list contains Spot bid status codes and explanations about what the code means for your Spot request:

- **az-group-constraint**

The Spot service cannot launch all the instances you requested in the same Availability Zone.

- **bad-parameters**

One or more of the parameters of your Spot request was invalid or malformed (for example, the AMI you specified may not exist). The bid status message will tell you which parameter is invalid.

- **canceled-before-fulfillment**

The user canceled the Spot request before it was fulfilled.

- **capacity-not-available**

There is no capacity available for the instances you requested.

- **capacity-oversubscribed**

The number of Spot requests with bid prices equal to or higher than your bid price exceeds the available capacity in this pool.

- **constraint-not-fulfillable**

The Spot request cannot be fulfilled because one or more constraints are invalid. For example, you provided an invalid *from* or *to* or an invalid Availability Zone. The bid status message will tell you which constraint was invalid.

- **fulfilled**

Your bid request was fulfilled, so your Spot request is active, and the Amazon EC2 Spot service is launching (or has launched) your Spot Instance.

- **instance-terminated-by-user (or spot-instance-terminated-by-user)**

You terminated a Spot instance that had been fulfilled, so the bid state becomes closed (unless it's a persistent bid), and the instance state is terminated.

Note

The status codes `instance-terminated-by-user` and `spot-instance-terminated-by-user` represent the same status. The code `spot-instance-terminated-by-user` will be deprecated.

- **instance-terminated-by-price**

The Spot price rose above your bid price. If your request is a persistent bid, the process—or *life cycle*—restarts and your bid will again be pending evaluation.

- **instance-terminated-capacity-oversubscribed**

Your instance was terminated because the number of Spot requests with bid prices equal to or higher than your bid price has exceeded the available capacity in this pool. This means that your instance was interrupted even though the Spot price may not have changed because your bid was at the Spot price.

Note

When there are several requests at a certain bid price and only some of them need to be terminated, the Spot service randomly selects the requests to be terminated.

- **instance-terminated-launch-group-constraint**

One of the instances in your launch group was terminated, so the launch group constraint is no longer fulfilled.

- **instance-terminated-no-capacity**

There is no longer any Spot capacity available for the instance.

- **launch-group-constraint**

The Spot service cannot launch all the instances you requested at the same time.

- **marked-for-termination**

Your Spot Instance is marked for termination.

- **not-scheduled-yet**

Your Spot request will not be evaluated until the scheduled date.

- **pending-evaluation**

As soon as you make your Spot Instance request, it goes into the pending-evaluation state while the system evaluates the parameters of your request.

- **pending-fulfillment**

Your Spot request is pending fulfillment while the system tries to provision your Spot Instance.

- **placement-group-constraint**

Your Spot request cannot be fulfilled yet because a new Spot Instance cannot currently be added to the placement group you specified.

- **price-too-low**

You have a bid request that cannot be fulfilled because the bid price is below the Spot price. In this case, no instance is launched, and your bid remains open.

- **request-canceled-instance-running**

You canceled your Spot request while the Spot instance was still running, so the request is canceled, but the instance remains running.

- **schedule-expired**

Your Spot request has expired because it was not fulfilled before the valid *to* date.

- **system-error**

There was an unexpected system error. If this is a recurring issue, please contact customer support for assistance.

Tagging Spot Instance Requests

To help categorize and manage your Spot Instance requests, you can tag them with metadata of your choice. You tag your Spot Instance requests in the same way that you tag other Amazon EC2 resources. Create a tag by specifying a key and value, and assigning the pair to the Spot Instance request. You use the [CreateTags](#) action or the [ec2-create-tags](#) command. For information about how to use tags, see [Tagging Your Amazon EC2 Resources \(p. 540\)](#).

The tags you create for your Spot Instance requests only apply to the requests. These tags don't propagate to the instances that the requests launch. To categorize the Spot Instances that are launched from your tagged request, you must create a separate set of tags for the instances using the same commands.

For example, you have Spot Instance request *sir-69047e11* and you want to label it with the tag *Spot=Test*. To do this, use the following command:

```
PROMPT> ec2-create-tags sir-69047e11 --tag "Spot=Test"
```

Amazon EC2 returns this information:

TAG	spot-instance-request	sir-69047e11	Spot	Test
-----	-----------------------	--------------	------	------

You can also confirm the tag information by using the [ec2-describe-tags](#) command.

When your request is fulfilled and a Spot Instance launches, you will see that the tag you used for your Spot Instance request is not applied to the Spot Instance. In the following example command, we are obtaining information about the Spot Instance *i-b8ca48d8* that was launched as a result of your Spot Instance request *sir-69047e11*, tagged *Spot=Test*.

```
PROMPT> ec2-describe-instances i-b8ca48d8
```

The call returns details about the instance with no tag information, showing that the tag for your Spot Instance request does not apply to the Spot Instance that the request launched. To tag your Spot Instance, use the following command:

```
PROMPT> ec2-create-tags i-b8ca48d8 --tag "SpotI=Test1"
```

Amazon EC2 returns this information:

TAG	instance	i-b8ca48d8	SpotI	Test1
-----	----------	------------	-------	-------

You can create similar tags using the API. For more information, see the API section of [Tagging Your Amazon EC2 Resources \(p. 540\)](#).

Protecting Your Spot Instance Data Against Interruptions

Customer demand for Amazon EC2 Spot Instances can vary significantly from moment to moment, and the availability of Spot Instances can also vary significantly depending on how much Amazon EC2 capacity is available. Therefore, it is always possible that your Spot instance will be interrupted if another user bids higher than you or if we have to reclaim spare capacity.

Note

No matter how high you bid, there is always a risk that your Spot Instance will be interrupted. We strongly recommend against bidding above the On-Demand price or using Spot for applications that cannot tolerate interruptions.

The following sections discuss how to plan for and mitigate the impact of interruptions.

- [Planning for Interruptions \(p. 160\)](#)
- [Persisting Your Root EBS Partition \(p. 160\)](#)

Planning for Interruptions

Topics

- [Quick Look: Managing Interruptions Video \(p. 160\)](#)

Because Spot Instances can terminate at any time, applications that run on Spot Instances must terminate cleanly. Although we attempt to cleanly terminate your instances, your application should be prepared to deal with an immediate shutdown.

To test your application, launch and terminate it as an On-Demand instance.

Quick Look: Managing Interruptions Video

The following video shows how some customers manage the interruption of their Spot instances. [How to Manage Spot Instance Interruptions](#)

For more information about strategies to manage interruptions, go to the following sections:

- [Launching Amazon Elastic MapReduce Job Flows with Spot Instances \(p. 177\)](#)
- [Using Auto Scaling to Get Notifications for Spot Instances \(p. 174\)](#)
- [Starting Clusters on Spot Instances \(p. 210\)](#)

Persisting Your Root EBS Partition

Amazon Elastic Block Store (Amazon EBS) can be an effective way to store data that you otherwise might lose when your Spot Instance terminates.

Important

Although Spot Instances can use Amazon EBS-backed AMIs, be aware that Spot Instances do not support the Stop/Start feature. In other words, you can't stop and start Spot Instances launched from an AMI that has an Amazon EBS root device. For information about Amazon EBS, see [Amazon Elastic Block Store \(Amazon EBS\)](#).

To set up the persistence of Spot Instance data, you map the Spot Instances that will be launched to an existing Amazon Elastic Block Store (Amazon EBS) snapshot. Set the *delete-on-termination* flag to *false*; this indicates that Amazon EC2 shouldn't delete the Amazon EBS volume when the spot instance terminates.

Let's walk through making an example Spot request with the following specifications:

- Bid price of \$0.500
- One instance of the m1.xlarge instance type
- Block device mapping to a snapshot that shouldn't be deleted when the Spot Instance is terminated

You can do this example using either the CLI or API tools. Using the CLI, your example request should look like this:

```
PROMPT> ec2-request-spot-instances -p 0.5 -t m1.xlarge -n 1 -b '/dev/sdb=snap-a123bcde:20:false' ami-8e1fece7
```

For more information, see:

- [Block Device Mapping \(p. 522\)](#)
- [ec2-request-spot-instances](#)
- [RequestSpotInstances](#)

Walkthroughs: Using Spot Instances with AWS Services

Topics

- [Managing Spot Instances with Auto Scaling \(p. 161\)](#)
- [Using CloudFormation Templates to Launch Spot Instances \(p. 176\)](#)
- [Launching Amazon Elastic MapReduce Job Flows with Spot Instances \(p. 177\)](#)
- [Launching Spot Instances in Amazon Virtual Private Cloud \(p. 178\)](#)

You can use AWS services with Spot Instances. In this section, we will show you how Amazon EC2 Spot Instances works with services, such as Auto Scaling, Elastic MapReduce, and Amazon Virtual Private Cloud (Amazon VPC).

Managing Spot Instances with Auto Scaling

Topics

- [Tools for Managing Auto Scaling with Spot Instances \(p. 162\)](#)
- [Launching Spot Instances with Auto Scaling \(p. 164\)](#)
- [Obtaining Information About the Instances Launched by Auto Scaling \(p. 167\)](#)
- [Updating the Bid Price for the Spot Instances \(p. 171\)](#)
- [Scheduling Spot Bid Requests \(p. 173\)](#)
- [Using Auto Scaling to Get Notifications for Spot Instances \(p. 174\)](#)

You can take advantage of Auto Scaling features to manage your Amazon EC2 Spot Instances. With Auto Scaling, you can scale up or down your capacity based on demand by setting up Auto Scaling to make Spot bids on your behalf.

In addition, you can use Auto Scaling's scheduling functionality for more granular control over when to bid and launch your Spot Instances. You also can use an Amazon Simple Notification Service (Amazon SNS)-backed Auto Scaling feature that sends notifications each time specified events—such as the launch or termination of instances—take place. Using Auto Scaling's scheduled actions, you can set up bids to expire at a set time.

When you use Auto Scaling with Spot Instances, there are some Auto Scaling tools that you have to use instead of the Spot Instance tools that might be familiar. There are also a number of Spot Instance options that you cannot use. Here's a summary of differences:

- **Setting your bid price.** When you use Auto Scaling to launch Spot Instances, you set your bid price in an Auto Scaling launch configuration.
- **Spot market price and your bid price.** This is the same as current behavior. If the market price for Spot Instances rises above your bid price for a running instance, Amazon EC2 will terminate your instance.
- **Changing your bid price.** If you want to change your bid price, you have to create a new launch configuration and associate it with your Auto Scaling group. You cannot update the existing launch configuration.
- **New bid price and running instances.** When you change your bid price by creating a new launch configuration, instances already launched will continue to run as long as the bid price for those running instances is higher than the current market price for Spot Instances.
- **Spot and Auto Scaling instance termination.** Amazon EC2 terminates a Spot Instance when the bid price for that instance falls below the Spot market price. Auto Scaling terminates instances based on a combination of criteria, including the launch configuration it is associated with, length of time in a billing hour the instance has been running, and the Availability Zone in which it is launched. For more information about instance termination in Auto Scaling, see [Auto Scaling Instance Termination](#).
- **Maintaining your Spot Instances.** When your instance is terminated, Auto Scaling will try to launch another instance to replace it in order to maintain your specified desired capacity. However, whether or not Auto Scaling successfully launches an instance depends on the bid price as compared to the Spot market price: If the bid price is higher than the Spot market price, then an instance will be launched; if the Spot market price is higher than the bid price, then no instance will be launched at that point.
- **Persistent bids.** Auto Scaling will continue submitting bids for you as long as you keep your Auto Scaling group and launch configuration. The Auto Scaling group specifies the desired number of instances you want maintained and the launch configuration specifies the bid price for your Spot Instances.

For more information about Auto Scaling, see [Auto Scaling Developer Guide](#). To get started quickly with Auto Scaling, see [Auto Scaling Getting Started Guide](#).

Tools for Managing Auto Scaling with Spot Instances

You will use the Auto Scaling command line interface (CLI) tools to use the Auto Scaling features with Spot Instances. Currently, the AWS Management Console does not have support to create and manage Auto Scaling objects. However, you can view most details about bids and Amazon EC2 instances launched by Auto Scaling on the AWS Management Console.

To use the Auto Scaling CLI tools, download and unzip the package, and set up the tools on your computer just as you set up your Amazon EC2 CLI tools. The Auto Scaling CLI tools and the Amazon EC2 CLI tools are shipped as different tools packages.

The following table lists resources that can help you get started with the Auto Scaling CLI tools. For more information, go to [Tools](#) in the *Auto Scaling Developer Guide*.

Resources for Using Auto Scaling CLI Tools

Task	Resource
Download the Auto Scaling command line interface tools	Auto Scaling Command Line Tool on the Amazon Web Services site.
Setup and Install Auto Scaling command line interface tools	Install the Command Line Interface in the <i>Auto Scaling Developer Guide</i> .
Read the readme installation instructions	<i>Install Drive\AutoScaling-n.n.nn.n\README</i>

To get help about Auto Scaling CLI commands

After you have installed the Auto Scaling CLI tools, you can get more information about the commands by querying the command line.

1. Open the Command Line.
2. Navigate to *Install Drive\AutoScaling-n.n.nn.n\bin*.
3. Type `as-cmd` and press **Enter**.

The command line returns a list of Auto Scaling CLI commands and a short description of what each command does.

For example, here are some of the Auto Scaling CLI commands we will use in this section.

Command	Description
<code>as-create-launch-config</code>	Creates a new launch configuration with specified attributes.
<code>as-create-auto-scaling-group</code>	Creates a new Auto Scaling group with specified name and other attributes.
<code>as-describe-auto-scaling-groups</code>	Describes the Auto Scaling groups, if the groups exist.
<code>as-describe-auto-scaling-instances</code>	Describes the Auto Scaling instances, if the instances exist.
<code>as-describe-scaling-activities</code>	Describes a set of activities or all activities belonging to a group.
<code>as-delete-auto-scaling-group</code>	Deletes the specified auto scaling group, if the group has no instances and no scaling activities are in progress.

For common conventions the documentation uses to represent command symbols, see [Document Conventions](#).

4. Type `as-command-name --help`.

The command line returns a description, syntax information, and examples of how to use the specified Auto Scaling CLI command.

For Amazon EC2 CLI tools, see the following:

- **Amazon EC2 CLI tools.** [Amazon Elastic Compute Cloud CLI Reference](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.
- **Amazon EC2 CLI tools specific to Spot Instances.** [Amazon EC2 Command Line Interface \(CLI\) Tools \(p. 135\)](#) in the *Amazon Elastic Compute Cloud User Guide*.
- **Other Amazon EC2 tools you can use with Spot instances.** [Getting Started with Spot Instances \(p. 133\)](#) in the *Amazon Elastic Compute Cloud User Guide*.

Launching Spot Instances with Auto Scaling

In this section, we will create an Auto Scaling launch configuration and an Auto Scaling group that launch Spot Instances. We will use the Auto Scaling command line interface (CLI) commands to create the launch configuration and the Auto Scaling group, and to verify and obtain information about them and the Amazon EC2 instances that they launch.

Prerequisites

If you're not familiar with how to create a launch configuration or an Auto Scaling group, we recommend that you go through the steps in the [Basic Auto Scaling Configuration](#) in the *Auto Scaling Developer Guide*. Use the basic scenario to get started with the infrastructure that you need in most Auto Scaling scenarios.

If you don't have the Auto Scaling CLI tools installed on your computer, you must install them to do this walkthrough. For information, see [Tools for Managing Auto Scaling with Spot Instances](#) in the *Amazon Elastic Compute Cloud User Guide*. You can also use the information in [Install the Command Line Interface](#) in the *Auto Scaling Developer Guide*.

In this scenario, we will perform the following tasks:

- [Step 1: Create a Launch Configuration \(p. 164\)](#)
- [Step 2: Create an Auto Scaling Group \(p. 165\)](#)

If you already have a launch configuration and Auto Scaling group, here are other related Spot Instance tasks that you can do using Auto Scaling:

- [Updating the Bid Price for the Spot Instances \(p. 171\)](#)
- [Scheduling Spot Bid Requests \(p. 173\)](#)
- [Using Auto Scaling to Get Notifications for Spot Instances \(p. 174\)](#)
- [Advanced Tasks \(p. 180\)](#)

For more information about Auto Scaling, see [What is Auto Scaling?](#) in the *Auto Scaling Developer Guide*. For information about scenarios using Auto Scaling, see [Using Auto Scaling](#) also in the *Auto Scaling Developer Guide*.

Step 1: Create a Launch Configuration

An Auto Scaling launch configuration is a template that contains the parameters necessary to launch new Amazon EC2 instances. You can attach only one launch configuration to an Auto Scaling group at a time. You have to create a launch configuration and then an Auto Scaling group in order to launch instances with Auto Scaling.

To place bids for Spot Instances, use the `as-create-launch-config` Auto Scaling CLI command with the `--spot-price` option. Specify the maximum price you want to pay for an instance. This price will be used by Auto Scaling to bid for your instances, but this price is not necessarily what you pay for the instances when they are launched. You will pay the Spot price. For example, you bid \$0.05 for m1.small instances. Your bid gets fulfilled if the current market price for m1.small Spot Instance is \$0.03, or any

other price below \$0.05, and you will be charged the current price of \$0.03 per hour. In fact, as long as your current bid is higher than the market price, your bid will be fulfilled and a Spot Instance will be launched for you.

You can get guidance on the price to bid by checking Spot price history. You can do this by using the AWS Management Console, CLI, or API. For more information, go to [Viewing Spot Price History \(p. 136\)](#).

The `as-create-launch-config` command takes the following arguments:

```
as-create-launch-config LaunchConfigurationName --image-id value --instance-type  
  value [--spot-price value] [--iam-instance-profile value] [--block-device-mapping  
  "key1=value1,key2=value2..." ] [--monitoring-enabled|--monitoring-disabled]  
  [--kernel value] [--key value] [--ramdisk value] [--group value[,value...]]  
  ] [--user-data value] [--user-data-file value] [General Options]
```

The only required options are the launch configuration name, image ID, and instance type. For this walkthrough, specify:

- Launch configuration name = `spotlc-5cents`

Note

When Auto Scaling launches instances, it does not distinguish the Spot Instances from the On-Demand instances. To help you identify which of your instances are Spot Instances, consider assigning a launch configuration name that includes *spot* and the bid price.

- Image ID = `ami-e565ba8c`

The Image ID identifies the Amazon Machine Image (AMI) that will be used to launch the instances. If you don't have an AMI, and you want to find a suitable one, see [Amazon Machine Images \(AMIs\)](#).

- Instance type = `m1.small`
- Spot price = \$0.05

This parameter is optional. If you want to launch Spot Instances, you must specify the Spot bid price that will be used to bid for Spot Instances. Spot Instance bid prices must be specified in US dollars.

Your command should look similar to the following example:

```
as-create-launch-config spotlc-5cents --image-id ami-e565ba8c --instance-type  
m1.small --spot-price "0.05"
```

You should get a confirmation like the following example:

```
OK-Created launch config
```

Step 2: Create an Auto Scaling Group

An Auto Scaling group is a collection of Amazon EC2 instances that shares similar characteristics and to which you want to apply certain scaling actions. You can use the Auto Scaling group to automatically scale the number of instances or maintain a fixed number of instances. You can attach only one launch configuration to an Auto Scaling group at a time. You have to create a launch configuration and then an Auto Scaling group in order to launch Amazon EC2 instances with Auto Scaling.

The `as-create-auto-scaling-group` command takes the following arguments:

```
as-create-auto-scaling-group AutoScalingGroupName --availability-zones  
  value[,value...] --launch-configuration value --max-size value --min-size value  
  [--default-cooldown value] [--desired-capacity value] [--grace-period value]
```

```
[--health-check-type value] [--load-balancers value[ , value]] [--placement-group value] [--vpc-zone-identifier value] [General Options]
```

This command requires that you specify a name for your Auto Scaling group, a launch configuration, one or more Availability Zones, a minimum group size, and a maximum group size. The Availability Zones you choose determine the physical location of your Auto Scaling instances. The minimum and maximum group size tells Auto Scaling the minimum and maximum number of instances the Auto Scaling group should have.

Desired capacity is an important component of the `as-create-auto-scaling-group` command. Although it is an optional parameter, desired capacity tells Auto Scaling the number of instances you want to run initially. To adjust the number of instances you want running in your Auto Scaling group, you change the value of `--desired-capacity`. If you don't specify `--desired-capacity`, its value is the same as minimum group size.

For more detailed information on the syntax of the `as-create-auto-scaling-group` command, see [Basic Auto Scaling Configuration](#) in the *Auto Scaling Developer Guide*. You can also get help information from the command line: Run the `as-create-auto-scaling-group --help`. For more information, go to [Resources for Using Auto Scaling CLI Tools \(p. 163\)](#).

For this walkthrough, specify these values for the command:

- Auto Scaling Group name = `spotasg`

Note

When Auto Scaling launches instances, it does not distinguish the Spot Instances from the On-Demand instances. To help you identify which of your instances are Spot Instances, consider assigning spot-specific names to the Auto Scaling group that launches Spot Instances.

- Launch configuration name = `spotlc-5cents`
- Availability Zone = `us-east-1a,us-east-1b`
- Max size = 5
- Min size = 1
- Desired capacity = 3

Your command should look like the following example:

```
as-create-auto-scaling-group spotasg --launch-configuration spotlc-5cents --availability-zones "us-east-1a,us-east-1b" --max-size 5 --min-size 1 --desired-capacity 3
```

You should get confirmation like the following example:

```
OK-Created AutoScalingGroup
```

What do you want to do next?

- [Obtaining Information About the Instances Launched by Auto Scaling \(p. 167\)](#)
- [Updating the Bid Price for the Spot Instances \(p. 171\)](#)
- [Scheduling Spot Bid Requests \(p. 173\)](#)
- [Using Auto Scaling to Get Notifications for Spot Instances \(p. 174\)](#)
- [Advanced Tasks \(p. 180\)](#)

For more information about Auto Scaling, see [What is Auto Scaling?](#) in the *Auto Scaling Developer Guide*. For information about scenarios using Auto Scaling, see [Using Auto Scaling](#) also in the *Auto Scaling Developer Guide*.

Obtaining Information About the Instances Launched by Auto Scaling

You can use the Auto Scaling CLI tools to obtain information about your launch configuration, Auto Scaling groups and Amazon EC2 instances launched by Auto Scaling.

In this section, we will use the following Auto Scaling CLI commands to get information about the Spot price bids and Spot Instances that Auto scaling makes and launches for you.

- **as-describe-scaling-activities**—You can use the information about Auto Scaling activities that this command returns to check the status of the bids submitted for you by Auto Scaling.
- **as-describe-auto-scaling-groups**—You can use the information about Auto Scaling groups that this command returns to confirm that Auto Scaling is launching your Spot Instances according to your specifications.

To check the status of the bids that Auto Scaling is making for you

The `as-describe-scaling-activities` command lists the activities that Auto Scaling performed for a specified Auto Scaling group.

This is the basic syntax:

```
as-describe-scaling-activities [ActivityIds [ActivityIds ...]]  
[--auto-scaling-group value] [--max-records value] [General Options]
```

Specifying the Auto Scaling group and the Activity ID are optional. If you don't specify the Auto Scaling group, the command will return all activities for all Auto Scaling groups associated with your account. If you specify the Auto Scaling group, only the activities for that Auto Scaling group will be listed.

In this scenario, we are using the `as-describe-scaling-activities` command to see state of your bid. Assume that there is only one Auto Scaling group (`spotasg`) and you want to list all activities.

1. Open a command line and navigate to the bin folder of your Auto Scaling CLI tools directory.
2. Type the command: `as-describe-scaling-activities --auto-scaling-group spotasg --headers`

The information you get should be similar to the following example.

ACTIVITY NAME	ACTIVITY-ID CODE	MESSAGE	END-TIME	GROUP-
ACTIVITY	31bcbb67-7f50-4b88-ae7e-e564a8c80a90	WaitingForSpotInstanceId Placed Spot instance request: sir-fc8a3014.		spotasg
	Waiting for instance(s)			
ACTIVITY	770bbeb5-407c-404c-a826-856f65db1c57	WaitingForSpotInstanceId Placed Spot instance request: sir-69101014.		spotasg
	Waiting for instance(s)			
ACTIVITY	597e4ebd-220e-42bc-8ac9-2bae4d20b8d7	Successful	2012-05-23T17:40:22Z	spotasg

In this response, you know that your bids were placed, one of the bids is successful, and Auto Scaling is waiting for the other two bids.

Note

If the `as-describe-scaling-activities` command returns a list that includes *Failed* activities, check the data you specified in the launch configuration. For example:

- The Amazon Machine Image (AMI) might not be valid anymore.
- The bid price specified in the launch configuration is lower than the Spot market price.

3. If you run the `as-describe-scaling-activities` command again later, you can be getting information that is similar to the following example:

ACTIVITY NAME	ACTIVITY-ID	END-TIME	GROUP-NAME
ACTIVITY CODE	90630906-b40f-41a6-967a-cd6534b2dfca	2012-06-01T02:32:15Z	spotasg
Successful	a1139948-ad0c-4600-9efe-9dab8ce23615	2012-06-01T00:48:02Z	spotasg
Successful	33001e70-6659-4494-a817-674d1b7a2f58	2012-06-01T02:31:11Z	spotasg
			Successful

The output shows that the listed activities were successful. Because we know that `spotasg` is an Auto Scaling group that uses a launch configuration with a spot bid price, you can assume that the activities represent the bids placed by Auto Scaling.

4. If you want to get more details about the activities and instances, use the `--show-xml` option of `as-describe-scaling-activities`. Enter the following command
- ```
as-describe-scaling-activities --auto-scaling-group spotasg --show-xml.
```

The information you get should be similar to the following example.

```
<DescribeScalingActivitiesResponse xmlns="http://autoscaling.&api-do
main;/doc/2011-01-01/">
 <DescribeScalingActivitiesResult>
 <NextToken>b5a3b43e-10c6-4b61-8e41-2756db1fb8f5</NextToken>
 <Activities>
 <member>
 <StatusCode>Successful</StatusCode>
 <Progress>0</Progress>
 <ActivityId>90630906-b40f-41a6-967a-cd6534b2dfca</ActivityId>
 <StartTime>2012-06-01T00:48:21.942Z</StartTime>
 <AutoScalingGroupName>spotasg</AutoScalingGroupName>
 <Cause>At 2012-06-01T00:48:21Z a difference between desired and ac
tual capacity changing the desired capacity, increasing the capacity from
2 to 3.</Cause>
 <Details>{}</Details>
 <Description>Launching a new EC2 instance: i-fe30d187</Description>

 <EndTime>2012-06-01T02:32:15Z</EndTime>
 </member>
 <member>
 <StatusCode>Successful</StatusCode>
 <Progress>0</Progress>
 <ActivityId>a1139948-ad0c-4600-9efe-9dab8ce23615</ActivityId>
 <StartTime>2012-06-01T00:47:51.293Z</StartTime>
 <AutoScalingGroupName>spotasg</AutoScalingGroupName>
 <Cause>At 2012-06-01T00:47:51Z an instance was taken out of service
in response to a system health-check.</Cause>
 </member>
 </Activities>
 </DescribeScalingActivitiesResult>
</DescribeScalingActivitiesResponse>
```

```
<Details>{}/</Details>
<Description>Terminating EC2 instance: i-88ce28f1</Description>
<EndTime>2012-06-01T00:48:02Z</EndTime>
</member>
<member>
<StatusCode>Successful</StatusCode>
<Progress>0</Progress>
<ActivityId>33001e70-6659-4494-a817-674d1b7a2f58</ActivityId>
<StartTime>2012-06-01T00:46:19.723Z</StartTime>
<AutoScalingGroupName>spotasg</AutoScalingGroupName>
<Cause>At 2012-06-01T00:46:19Z a difference between desired and ac-
tual capacity changing the desired capacity, increasing the capacity from
2 to 3.</Cause>
<Details>{}/</Details>
<Description>Launching a new EC2 instance: i-2c30d155</Description>

<EndTime>2012-06-01T02:31:11Z</EndTime>
</member>
...
</Activities>
</DescribeScalingActivitiesResult>
<ResponseMetadata>
<RequestId>d02af4bc-ad8f-11e1-85db-83e1968c7d8d</RequestId>
</ResponseMetadata>
</DescribeScalingActivitiesResponse>
```

The XML output shows more detail about the Spot and Auto Scaling activity.

- Cause: At 2012-06-01T00:48:21Z a difference between desired and actual capacity changing the desired capacity, increasing the capacity from 2 to 3. Description: Launching a new EC2 instance: i-2c30d155

If an instance is terminated and the number of instances falls below the desired capacity, Auto Scaling will launch a new instance so that the total number of your running instances rises back to the level specified for desired capacity.

- Cause: At 2012-06-01T00:47:51Z an instance was taken out of service in response to a system health-check. Description: Terminating EC2 instance: i-88ce28f1

Auto Scaling maintains the desired number of instances by monitoring the health status of the instances in the Auto Scaling group. When Auto Scaling receives notification that an instance is *unhealthy* or terminated, Auto Scaling launches another instance to take the place of the unhealthy instance. For information, see [Configure Health Checks for Your Auto Scaling group](#) in the *Auto Scaling Developer Guide*.

**Note**

Auto Scaling provides the cause of instance termination that is not the result of a scaling activity. This includes instances that have been terminated because the Spot market price exceeded their bid price.

When Auto Scaling attempts to replace terminated instances resulting from the Spot market price rising above the instances' bid price, Auto Scaling will place the bid specified in the current launch configuration and attempt to launch another instance to maintain the desired capacity.

## To confirm that Auto Scaling is launching your Spot Instances according to your specifications

Use `as-describe-auto-scaling-groups`. The command will show details about the group and instances launched. For information about the `as-describe-auto-scaling-groups` command, see [Verify Auto Scaling Group Creation](#) in the *Auto Scaling Developer Guide*.

1. Open a command line and navigate to the bin folder of your Auto Scaling CLI tools directory.
2. Type the command:`as-describe-auto-scaling-groups spotasg --headers`

**Note**

The `--headers` option supplies the column name so you know what data is being returned.

The information you get should be similar to the following example.

| AUTO-SCALING-GROUP | GROUP-NAME  | LAUNCH-CONFIG     | AVAILABILITY-ZONES    |
|--------------------|-------------|-------------------|-----------------------|
| MIN-SIZE           | MAX-SIZE    | DESIRED-CAPACITY  |                       |
| AUTO-SCALING-GROUP | spotasg     | spotlc-5cents     | us-east-1b,us-east-1a |
| 1                  | 5           | 3                 |                       |
| INSTANCE           | INSTANCE-ID | AVAILABILITY-ZONE | STATE                 |
| INSTANCE           | i-2c30d155  | us-east-1a        | InService             |
| INSTANCE           | i-fe30d187  | us-east-1a        | InService             |
| INSTANCE           | i-c630d1bf  | us-east-1a        | InService             |

You can see that Auto Scaling launched 3 instances in us-east-1a, as you specified, and they are all running.

3. Additionally, you can find out details about the Spot Instances launched for you by Auto Scaling, by using the `as-describe-auto-scaling-instances` command.

This is the basic syntax:

```
as-describe-auto-scaling-instances [InstanceIds [InstanceIds ...]]
[--max-records value] [General Options]
```

Specifying `InstanceIds` is optional. If you specify it, the command will return information about the instance, if it exists. If you don't specify `InstanceIds`, the command returns information about all instances associated with your Auto Scaling account.

In this walkthrough, we are assuming that you created one launch configuration and Auto Scaling group, and you want to find out details about all your Spot Instances.

Your command should look like the following example:

```
as-describe-auto-scaling-instances --headers
```

The information you get should be similar to the following example:

| INSTANCE      | INSTANCE-ID   | GROUP-NAME | AVAILABILITY-ZONE | STATE     | STATUS  |
|---------------|---------------|------------|-------------------|-----------|---------|
| LAUNCH-CONFIG |               |            |                   |           |         |
| INSTANCE      | i-2c30d155    | spotasg    | us-east-1a        | InService | HEALTHY |
|               | spotlc-5cents |            |                   |           |         |
| INSTANCE      | i-c630d1bf    | spotasg    | us-east-1a        | InService | HEALTHY |
|               | spotlc-5cents |            |                   |           |         |
| INSTANCE      | i-fe30d187    | spotasg    | us-east-1a        | InService | HEALTHY |
|               | spotlc-5cents |            |                   |           |         |

**What do you want to do next?**

- [Updating the Bid Price for the Spot Instances \(p. 171\)](#)
- [Scheduling Spot Bid Requests \(p. 173\)](#)
- [Using Auto Scaling to Get Notifications for Spot Instances \(p. 174\)](#)
- [Advanced Tasks \(p. 180\)](#)

## Updating the Bid Price for the Spot Instances

Auto Scaling launch configurations cannot be changed. If you want to modify your bid price for Spot Instances, you must create a new launch configuration.

If, for example, you want to launch a set of Spot Instances that have a higher likelihood of running uninterrupted for a long time, you can use a higher bid price. To do this, you must create a new launch configuration, using the same procedure that you followed earlier in this walkthrough. (For more information, go to [Step 1: Create a Launch Configuration \(p. 164\)](#).)

Specify the following values:

- Launch configuration name = spotlc-7cents
- Image ID = ami-e565ba8c

**Note**

If you don't have an AMI, and you want to find a suitable one, see [Amazon Machine Images \(AMIs\)](#).

- Instance type = m1.small
- Spot price = \$0.07

Your command should look similar to the following example:

```
as-create-launch-config spotlc-7cents --image-id ami-e565ba8c --instance-type m1.small --spot-price "0.07"
```

You should get a confirmation like the following example:

```
OK-Created launch config
```

After you have created the new launch configuration successfully, create a new Auto Scaling group specifying the new launch configuration.

Your command should look similar to the following example:

```
as-create-auto-scaling-group spotasg-7cents --launch-configuration spotlc-7cents --availability-zones "us-east-1a,us-east-1b" --max-size 5 --min-size 10 --desired-capacity 3
```

You should get a confirmation like the following example:

```
OK-Created AutoScalingGroup
```

You can view the status of your Spot bid and a list of the bids that Auto Scaling placed for you by running `as-describe-scaling-activities` soon after you create your Auto Scaling group.

Your command should look similar to the following example:

```
as-describe-scaling-activities --headers
```

If not all your bids are fulfilled, you will get information that looks like the following example:

| ACTIVITY                 | ACTIVITY-ID                           | END-TIME                                    | GROUP-   |
|--------------------------|---------------------------------------|---------------------------------------------|----------|
| NAME                     | CODE                                  | MESSAGE                                     |          |
| ACTIVITY                 | 5879cc50-1e40-4539-a754-1cb084f1aecd  |                                             | spotasg- |
| 7cents                   | WaitingForSpotInstanceId              | Placed Spot instance request: sir-93828812. |          |
| Waiting for instance(s)  |                                       |                                             |          |
| ACTIVITY                 | 777fbe1b-7a24-4aa-f-b7a9-d368d0511878 |                                             | spotasg- |
| 7cents                   | WaitingForSpotInstanceId              | Placed Spot instance request: sir-016cf812. |          |
| Waiting for instance(s)  |                                       |                                             |          |
| ACTIVITY                 | f4b00f81-eaea-4421-80b4-a2e3a35cc782  |                                             | spotasg- |
| 7cents                   | WaitingForSpotInstanceId              | Placed Spot instance request: sir-cf60ea12. |          |
| Waiting for instance(s)  |                                       |                                             |          |
| ACTIVITY                 | 31bcbb67-7f50-4b88-ae7e-e564a8c80a90  |                                             | spotasg  |
| WaitingForSpotInstanceId |                                       | Placed Spot instance request: sir-fc8a3014. |          |
| Waiting for instance(s)  |                                       |                                             |          |
| ACTIVITY                 | 770bbeb5-407c-404c-a826-856f65db1c57  |                                             | spotasg  |
| WaitingForSpotInstanceId |                                       | Placed Spot instance request: sir-69101014. |          |
| Waiting for instance(s)  |                                       |                                             |          |
| ACTIVITY                 | 597e4ebd-220e-42bc-8ac9-2bae4d20b8d7  | 2012-05-23T17:40:22Z                        | spotasg  |
| Successful               |                                       |                                             |          |
| ACTIVITY                 | eca158b4-a6f9-4ec5-a813-78d42c1738e2  | 2012-05-23T17:40:22Z                        | spotasg  |
| Successful               |                                       |                                             |          |
| ACTIVITY                 | 1a5bd6c6-0b0a-4917-8cf0-eee1044a179f  | 2012-05-23T17:22:19Z                        | spotasg  |
| Successful               |                                       |                                             |          |
| ACTIVITY                 | c285bf16-d2c4-4ae8-acad-7450655facb5  | 2012-05-23T17:22:19Z                        | spotasg  |
| Successful               |                                       |                                             |          |
| ACTIVITY                 | 127e3608-5911-4111-906e-31fb16d43f2e  | 2012-05-23T15:38:06Z                        | spotasg  |
| Successful               |                                       |                                             |          |
| ACTIVITY                 | bfb548ad-8bc7-4a78-a7db-3b41f73501fc  | 2012-05-23T15:38:07Z                        | spotasg  |
| Successful               |                                       |                                             |          |
| ACTIVITY                 | 82d2b9bb-3d64-46d9-99b6-054a9ecf5ac2  | 2012-05-23T15:30:28Z                        | spotasg  |
| Successful               |                                       |                                             |          |
| ACTIVITY                 | 95b7589b-f8ac-49bc-8c83-514bf664b4ee  | 2012-05-23T15:30:28Z                        | spotasg  |
| Successful               |                                       |                                             |          |
| ACTIVITY                 | 57bbf77a-99d6-4d94-a6db-76b2307fb9de  | 2012-05-23T15:16:34Z                        | spotasg  |
| Successful               |                                       |                                             |          |
| ACTIVITY                 | cdef758e-0be2-416e-b402-0ef521861039  | 2012-05-23T15:16:17Z                        | spotasg  |
| Successful               |                                       |                                             |          |
| ACTIVITY                 | d7e0a3ed-7067-4583-8a87-1561b3de2aed  | 2012-05-23T14:51:46Z                        | spotasg  |

```
Successful

ACTIVITY da5471ab-482c-4680-b430-99e4173d2bd7 2012-05-23T14:52:48Z spotasg
Successful

ACTIVITY 78701f3d-a747-46e1-8b0f-8aff22834f46 2012-05-23T14:38:17Z spotasg
Successful

ACTIVITY 274d4772-3614-4f5c-8858-026b33635be3 2012-05-23T14:38:16Z spotasg
Successful

ACTIVITY 1024abb2-bf84-4fae-b717-a398bac91c4f 2012-05-23T14:22:39Z spotasg
Successful
```

Bids are represented by values such as `sir-93828812` and `sir-016cf812`.

When you create a new launch configuration that sets a new bid price for Spot Instances, and you have Spot Instances already running based on a different price, these instances will continue running and will only be terminated if the Spot market price goes above the bid price on which it was based.

#### What do you want to do next?

- [Scheduling Spot Bid Requests \(p. 173\)](#)
- [Using Auto Scaling to Get Notifications for Spot Instances \(p. 174\)](#)
- [Advanced Tasks \(p. 180\)](#)

## Scheduling Spot Bid Requests

You can set up Auto Scaling to launch a certain number of instances at a specific time. This capability is useful if, for example, you want to take advantage of a window of time when prices historically are lower, or you want to terminate Spot Instances at a specific time.

We will use the Auto Scaling CLI command `as-put-scheduled-update-group-action` to set up a schedule. This is the basic syntax:

```
as-put-scheduled-update-group-action ScheduledActionName --auto-scaling-group
value [--desired-capacity value] [--end-time value] [--max-size value] [--min-size
value] [--recurrence value] [--start-time value] [--time value] [General Options]
```

In this scenario, use the following values:

- Scheduled action name: `as-spotbid-schedule`
- Auto Scaling group: `spotasg`
- Start time: `2012-05-15T19:10:00Z`
- End time: `2012-05-15T19:15:00Z`
- Desired capacity: `20`

Your command should look similar to the following example:

```
as-put-scheduled-update-group-action as-spotbid-schedule --auto-scaling-group
spotasg --desired-capacity 20 --start-time 2012-05-15T19:10:00Z --end-time 2012-
05-15T19:15:00Z
```

You should get a confirmation like the following example:

```
OK-Put Scheduled Update Group Action
```

To check your scheduled action, run `aws describe-scheduled-actions`.

You will get information similar to the following example:

```
UPDATE-GROUP-ACTION spotasg as-spotbid-schedule 2012-05-15T19:10:00Z 20
```

#### What do you want to do next?

- [Obtaining Information About the Instances Launched by Auto Scaling \(p. 167\)](#)
- [Updating the Bid Price for the Spot Instances \(p. 171\)](#)
- [Using Auto Scaling to Get Notifications for Spot Instances \(p. 174\)](#)
- [Advanced Tasks \(p. 180\)](#)

## Using Auto Scaling to Get Notifications for Spot Instances

You can set up Auto Scaling to send notifications to you as instances are terminated and launched. When the Spot market price goes up and your bid price falls below it, Amazon EC2 terminates the instances that were launched based on that bid price. If your Spot Instances are terminated, Auto Scaling will try to submit your bid and launch replacement instances and to ensure the capacity you specified for your Auto Scaling group. You can set up Auto Scaling to notify you when instance launch or termination events occur.

There are two ways to get notifications for Spot Instances:

- Auto Scaling
- AWS SDK Sample Code Tool

#### Spot Notifications Sample in AWS Java SDK

You can also use the AWS Java SDK to develop applications that monitor your Spot Instance usage in ways that are customized to your use case. The Spot Notifications sample application is a Java application that demonstrates techniques for monitoring Amazon EC2 instance status, Spot Instance requests, and Spot price fluctuations. The application is documented and freely available for download at [How to Track Spot Instance Activity with the Spot-Notifications Sample Application](#). You are free to adapt the application for your own purposes, or use it as a guide in developing your own application for monitoring Spot Instances. For more information, go to the [AWS SDK for Java](#).

#### Configuring Auto Scaling groups to send notifications about your Spot Instances

In this portion of the walkthrough, you learn how to set up Amazon SNS to send email notifications.

To do this, you need the following:

An *Amazon Resource Name (ARN)*. You generate an ARN when you create an Amazon Simple Notification Service (Amazon SNS) topic. A topic is a communication channel to send messages and subscribe to notifications. It provides an access point for publishers and subscribers to communicate with each other. An endpoint, such as an email address, must be subscribed to the topic so the endpoint can receive messages published to the topic. To create a topic, go to [Create a Topic](#) in the *Amazon Simple Notification Service Getting Started Guide*.

An *Auto Scaling Group*. Use the Auto Scaling group that you created earlier in this walkthrough.

A *notification configuration*. You configure an Auto Scaling group to send notifications when specified events take place by calling the `as-put-notification-configuration` CLI command or the `PutNotificationConfiguration` API action. We will discuss the steps in setting up a notification configuration later in this section. For more information about the command, go to [PutNotificationConfiguration](#) in the *Auto Scaling API Reference*.

A *list of Auto Scaling notification types*. These notification types are the events that will cause the notification to be sent.

### Set Up Notifications Using Auto Scaling

1. After you've created your Amazon SNS topic and you have the ARN, you are ready to set up the notification configuration. (To create a topic, go to [Create a Topic](#) in the *Amazon Simple Notification Service Getting Started Guide*.)

To configure your Auto Scaling group to send notifications when specified events take place, use the `as-put-notification-configuration` CLI command.

The `as-put-notification-configuration` command takes the following arguments:

```
as-put-notification-configuration AutoScalingGroupName --notification-types
value --topic-arn topic-ARN [General Options]
```

You need to specify the Auto Scaling group name, the ARN, and the notification types.

For this example, specify:

- Auto Scaling group name: `spotasg`

Specify the Auto Scaling group that you want to get notifications about. In this walkthrough, you want Auto Scaling to notify you when instances are launched and terminated for the `spotasg` Auto Scaling group.

- ARN:`arn:placeholder:MyTopic`

#### Note

ARNs are unique identifiers for Amazon Web Services (AWS) resources. Replace the ARN placeholder with your ARN.

- Notification types: `autoscaling:EC2_Instance_Launch`,  
`autoscaling:EC2_Instance_Terminate`

The notification types are the events you want Auto Scaling to send you e-mail about.

Open a command prompt and enter the `as-put-notification-configuration` command.

```
as-put-notification-configuration spotasg--topic-arn arn:placeholder:MyTopic
--notification-types autoscaling:EC2_INSTANCE_LAUNCH, autoscaling:EC2_INSTANCE_TERMINATE
```

Auto Scaling returns the following:

```
OK-Put Notification Configuration
```

You now have a notification configuration that sends a notification to the endpoint subscribed in the `arn:placeholder:MyTopic` ARN whenever instances are launched or terminated.

2. Verify the notification configuration.

To verify the notification actions associated with your Auto Scaling group when specified events take place, use `as-describe-notification-configurations`.

The `as-describe-notification-configurations` command takes the following arguments:

```
as-describe-notification-configurations [--auto-scaling-groups
value[,value...]] [--maxrecords value] [General Options]
```

If you specify the Auto Scaling group, this command returns a full list of all notification configuration for the Auto Scaling group listed. If you don't provide an Auto Scaling group name, the service returns the full details of all Auto Scaling groups. The command also returns a token if there are more pages to retrieve. To get the next page, call this action again with the returned token as the `next-token` argument. For this walkthrough, specify:

- Auto Scaling group name: `spotasg`

Open a command prompt and enter the `as-describe-notification-configurations` command.

```
as-describe-notification-configurations --auto-scaling-groups spotasg -headers
```

Auto Scaling returns the following:

```
NOTIFICATION-CONFIG GROUP-NAME TOPIC-ARN NOTIFICATION-TYPE-NAME
NOTIFICATION-CONFIG spotasg arn:placeholder:spotasg autoscaling:EC2_INSTANCE_LAUNCH
NOTIFICATION-CONFIG spotasg arn:placeholder:spotasg autoscaling:EC2_INSTANCE_TERMINATE
```

You have confirmed that you have a notification configuration set up for the `spotasg` Auto Scaling group.

When Auto Scaling launches instances to reach or maintain desired capacity, as specified in your Auto Scaling group, SNS sends a notification to your email address with information about the instances. You can check your email Inbox for this information, then run `as-describe-auto-scaling-group` to get information about current instances in the Auto Scaling group and confirm that the instances listed in your email actually exist.

#### What do you want to do next?

- [Obtaining Information About the Instances Launched by Auto Scaling \(p. 167\)](#)
- [Updating the Bid Price for the Spot Instances \(p. 171\)](#)
- [Scheduling Spot Bid Requests \(p. 173\)](#)
- [Advanced Tasks \(p. 180\)](#)

## Using CloudFormation Templates to Launch Spot Instances

You can use AWS CloudFormation templates to launch Spot Instances and the AWS resources you need for an application. Templates are text files describing a collection, called *stack*, of AWS resources that you want to deploy as a group. For more information about AWS CloudFormation templates, see [Learn Template Basics](#). For more information about using AWS CloudFormation, see the [AWS CloudFormation User Guide](#).

You can write your own template from scratch, or you can use one of the example templates from the [AWS CloudFormation Sample Template Library](#). The following templates in AWS CloudFormation utilize Spot Instances:

- Asynchronous Queue-Based Processing

This template creates an auto-scaled worker that monitors work (messages) in a queue. The application is auto-scaled based on the amount of work in the queue. When there is work, Auto Scaling scales up; when there is no work, Auto Scaling scales down. Each message contains a command/script to run, an input file location, and an output location for the results.

To download this template, go to [Asynchronous Queue-Based Processing](#).

- Bees with Machine Guns

This template creates a load balancer, a controller, and the instances behind the load balancer; fires up and triggers an attack; and stores logs on Amazon Simple Storage Service (Amazon S3) and then shuts down (if enabled). For this template, instances use the Amazon Linux AMI, and the setup requires that you already have an AWS Identity and Access Management (IAM) SSL certification. You can modify this template to remove the SSL dependencies.

**Note**

To launch this template, you need a new SSH private key that you create specifically for this task. You'll have to log in to the instance created by the template and provide this private key information.

To download this template, go to [Bees with Machine Guns](#).

- Grid Computing for Spot Instances

This template uses Star Cluster to launch and bootstrap a cluster of Amazon EC2 instances for high performance computational tasks using Spot pricing. You only need to provide the number of instances, instance size, and Spot price that you want to use.

To download this template, go to [Grid Computing for Spot Instances](#).

You can create stacks from these templates by using the AWS Management Console, the AWS CloudFormation command line interface tools, or the AWS CloudFormation API.

Before you use these templates, you must:

1. Have an AWS account and sign up for AWS CloudFormation.
2. Decide on the template to use.
3. Enter the parameters required for the stack.
4. Create the stack.

To get started with AWS CloudFormation, [Getting Started with AWS CloudFormation](#).

For more information about using AWS CloudFormation, see the [AWS CloudFormation User Guide](#).

## Launching Amazon Elastic MapReduce Job Flows with Spot Instances

You can launch an Amazon Elastic MapReduce job flow in Spot Instances. Amazon Elastic MapReduce is a data analysis tool that simplifies the set up and management of a computer cluster, the source data, and the computational tools that help you implement sophisticated data processing jobs quickly. For more information, see [What is Amazon EMR?](#).

In addition, you can use Spot Instances with Amazon EMR clusters and AWS Data Pipeline. For information, see [Amazon EC2 Spot Instances with Amazon EMR Clusters and AWS Data Pipeline](#).

## Quick Look: Using Spot Instances with Amazon Elastic MapReduce Video

The following video describes how Spot Instances work in Amazon EMR and walks you through the process of launching a job flow on Spot Instances from the AWS Management Console: [Using Spot Instances with Amazon ElasticMapReduce](#)

## Launching Spot Instances in Amazon Virtual Private Cloud

Amazon Virtual Private Cloud (Amazon VPC) lets you define a virtual networking environment in a private, isolated section of the Amazon Web Services (AWS) cloud. You have complete control of this virtual network and you can use advanced security features and network access control at the instance level and subnet level. For more information about Amazon VPC, see the [Amazon Virtual Private Cloud User Guide](#).

If you want to take advantage of the features of Amazon VPC when you use Spot Instances, specify in your Spot request that your instances are to be launched in Amazon VPC. To ensure that your Spot Instances are launched in Amazon VPC:

- Determine if your AWS account supports a default VPC. For more information, see [Detecting Your Supported Platforms and Whether You Have a Default VPC](#).
- If your AWS account supports a default VPC, and
  - You want to launch your Spot Instances in that VPC

You don't have to perform any additional steps when you request Spot Instances. The **Request Spot Instances** wizard selects the subnet ID of your default VPC in which to launch the instance.

- You want to use a different VPC for the Spot Instances

Create a new Amazon VPC and specify the subnet ID of the VPC that you just created.

**Note**

Support for default VPC is available in select regions, and will be available in all regions soon. For more information, see [Default VPC Basics](#).

- If your AWS account does not support a default VPC, create an Amazon VPC and specify the subnet of the VPC that you just created.

This topic discusses what you need to know if you want your Spot Instances to launch in an Amazon VPC and your account does not support default VPCs, or your account supports default VPCs but you want your Spot Instances to launch in a VPC that is not the default.

For more information about setting up an Amazon VPC, see [Getting Started with Amazon VPC](#) in the [Amazon Virtual Private Cloud Getting Started Guide](#).

## Quick Look: Launching Spot Instances in Amazon VPC Video

The following video shows how to launch your first Spot Instance in the Amazon Virtual Private Cloud (Amazon VPC) using the AWS Management Console. This video includes instructions for creating your Amazon VPC subnet, placing a bid, determining when the instance is fulfilled, and canceling the instance. [Launching Spot Instances in Amazon VPC Video](#)

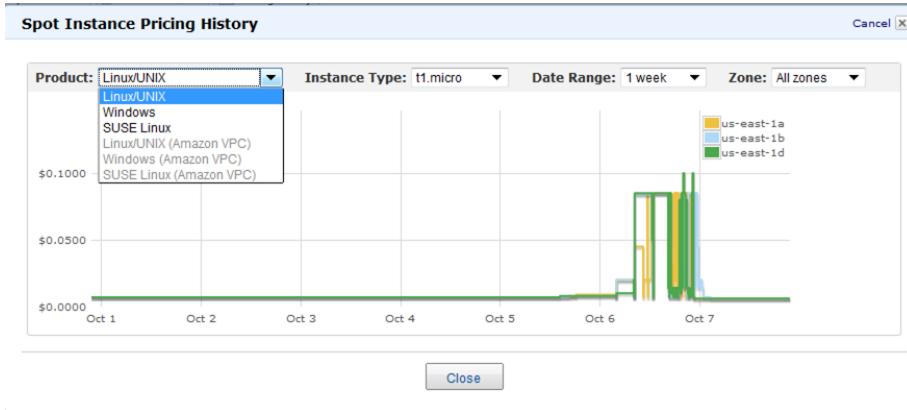
The process for making a Spot Instance request that launches in Amazon VPC is the same as the process you follow when you make a Spot Instance request in a non-VPC portion of Amazon EC2. The main differences are that you:

- Base your Spot price bid on Spot price history of Spot Instances in VPCs.

When you use the `DescribeSpotPriceHistory` action or the `ec2-describe-spot-price-history` command, add *Amazon VPC* to the *product-description* filter. For example:

```
PROMPT> ec2-describe-spot-price-history -s '2011-09-08T00:00:00Z' -t m1.xlarge
-d "Linux/UNIX (Amazon VPC)"
```

Using the AWS Management Console, check the **Spot Price History** page to see the Spot pricing history for Amazon EC2 instances running in both Amazon EC2 and Amazon VPC.



- Specify the VPC subnet in which you want to launch your Spot Instance.

When you use the `RequestSpotInstances` action or the `ec2-request-spot-instances` command, specify the ID of the Amazon VPC subnet in which you want to launch the Spot Instance.

```
PROMPT> ec2-request-spot-instances ami-8elfece7 -t m1.xlarge -p '$0.01' -n 5
-r 'one-time' -s 'subnet-baab943d3'
```

When you launch the **Request Instances Wizard** from the Spot Instance page of the AWS Management Console, select a subnet after specifying that you're launching the Spot Instance into a VPC.

The figure is a screenshot of the AWS Management Console's Request Instances Wizard. It shows the 'Request Spot Instances' step. The 'Number of Instances' is set to 1 and the 'Instance Type' is 'Extra Large (m1.xlarge, 15 GB)'. Under 'Launch Into:', the 'VPC' option is selected. In the 'Subnet' dropdown, three subnets are listed: 'subnet-f67849f (10.0.1.0/24) us-east-1d', 'subnet-f67849f (10.0.1.0/24) us-east-1d', and 'subnet-c5784ac (10.0.1.0/24) us-east-1a'. The 'us-east-1d' subnet is currently selected. At the bottom, there are 'Back' and 'Continue' buttons.

#### Note

The **Request Spot Instances** page shows the **VPC** option only if you have created a VPC and subnet before making the Spot Instance request.

For more information about using Amazon VPC, see the [Amazon Virtual Private Cloud User Guide](#).

## Advanced Tasks

Now that you have created Spot Instance requests and worked with Spot Instances, this section discusses some advanced tasks.

- [Subscribe to Your Spot Instance Data Feed \(p. 180\)](#)
- [Programming Spot with AWS Java SDK \(p. 183\)](#)
- [Starting Clusters on Spot Instances \(p. 210\)](#)

If you think we should add other advanced tasks to this section, let us know at [spot-instance-feedback@amazon.com](mailto:spot-instance-feedback@amazon.com).

### Subscribe to Your Spot Instance Data Feed

If you want to monitor your Spot Instance usage, you can subscribe to your Spot Instance data feed, which stores usage logs in Amazon Simple Storage Service (Amazon S3).

This section describes the data feed content and how to create the data feed for Spot Instances. You can create one data feed per account.

#### Spot Instance Data Feed Overview

To help you understand the charges for your Spot Instances, Amazon EC2 provides access to a data feed that details your Spot Instance usage and pricing. This data feed is sent to the Amazon S3 bucket of your choice.

To have a data feed delivered to an Amazon S3 bucket, you need to create a Spot Instances data feed subscription using the Amazon EC2 API. When you create this subscription, you can specify an Amazon S3 bucket to deliver the data feed files to, and a filename prefix to use to avoid collisions.

#### Data Feed Filename and Format

The Spot Instance data feed filename uses the following format (with the date and hour in UTC):

```
{Bucket}.s3.amazonaws.com/{Optional Prefix}/{AWS Account ID}.{YYYY}-{MM}-{DD}-
{HH}.{n}.{Unique ID}.gz
```

For example, if your bucket name is `myawsbucket`, and you name your prefix `myprefix`, your filenames look similar to this:

```
myawsbucket.s3.amazonaws.com/myprefix/111122223333.2010-03-17-20.001.pwBdGTJG.gz
```

Data feed files arrive in your bucket typically once an hour and each hour of usage is typically covered in a single data file. These files are compressed (gzip) before delivery into your bucket. We can write multiple files for a given hour of usage where files are very large (for example, when file contents for the hour exceed 50 MB before compression).

#### Note

If you don't have a Spot Instance running during a certain hour, you won't receive a data feed file for that hour.

The Spot Instance data feed files are tab-delimited. Each line in the data file corresponds to one instance-hour. Each line contains the fields listed in the following table.

| Field       | Description                                                                                                                                                                                                                                          |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Timestamp   | The timestamp used to determine the price charged for this instance-hour.                                                                                                                                                                            |
| UsageType   | Indicates the type of usage and instance type being charged for. For m1.small Spot Instances, this field is set to "SpotUsage." For all other instance types, this field is set to "SpotUsage:{instance-type},", for example, "SpotUsage:c1.medium." |
| Operation   | Indicates the product being charged for. For Linux/UNIX Spot Instances, this field is set to "RunInstances." For Microsoft Windows, this field is set to "RunInstances:0002." Spot usage is grouped according to Availability Zone.                  |
| InstanceID  | The instance ID for the Spot Instance that generated this instance-hour.                                                                                                                                                                             |
| MyBidID     | The Spot Instance request ID for the request that generated this instance-hour.                                                                                                                                                                      |
| MyMaxPrice  | The maximum price specified for this Spot Instance request.                                                                                                                                                                                          |
| MarketPrice | The Spot price at the time specified in the Timestamp field.                                                                                                                                                                                         |
| Charge      | The price charged for this instance-hour.                                                                                                                                                                                                            |
| Version     | The version included in the data feed filename for this record.                                                                                                                                                                                      |

### Preparing Amazon S3 for Data Feeds

When you subscribe to data feeds, you tell Amazon EC2 which bucket you want to store the data feed file in. Before you subscribe to the data feed, consider the following when choosing your S3 bucket:

- You must have Amazon S3 FULL\_CONTROL permission on the bucket you provide.  
If you're the bucket owner, you have this permission by default. If you're not the bucket owner, the bucket owner must grant your AWS account FULL\_CONTROL permission.
- When you create your data feed subscription, Amazon EC2 updates the designated bucket's ACL to allow read and write permissions for the AWS data feeds account.
- Each data feed file has its own ACL (separate from the bucket's ACL).

The bucket owner has FULL\_CONTROL permission for the data files. The data feed account has read and write permission.

- Removing the permissions for the data feed account does not disable the data feed.

If you remove those permissions but don't disable the data feed (which you do with the control API), we reinstate those permissions the next time the data feeds account needs to write a data file to your bucket.

- If you delete your data feed subscription, Amazon EC2 doesn't remove the read/write permissions for the data feed account on either the bucket or the data files.

You must perform remove the read/write permissions yourself.

## Subscribe to Your Spot Instance Data Feed

### Amazon EC2 Command Line Interface (CLI) Tools

#### To subscribe to your Spot Instance data feed

- Enter the following command:

```
PROMPT> ec2-create-spot-datafeed-subscription --bucket myawsbucket [--prefix prefix]
```

Amazon EC2 returns output similar to the following:

|                          |              |             |        |
|--------------------------|--------------|-------------|--------|
| SPOTDATAFEEDSUBSCRIPTION | 111122223333 | myawsbucket | prefix |
| Active                   |              |             |        |

## API

### To subscribe to your Spot Instance data feed

- Construct the following Query request.

```
https://ec2.amazonaws.com/
?Action=CreateSpotDatafeedSubscription
&Bucket=myawsbucket
&Prefix=my-spot-subscription
&...auth parameters...
```

Following is an example response.

```
<CreateSpotDatafeedSubscriptionResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
 <requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId>
 <spotDatafeedSubscription>
 <ownerId>9999988887777</ownerId>
 <bucket>myawsbucket</bucket>
 <prefix>my-spot-subscription</prefix>
 <state>Active</state>
 <fault>/</fault>
 </spotDatafeedSubscription>
</CreateSpotDatafeedSubscriptionResponse>
```

## Delete a Spot Instance Data Feed

### Command Line Tools

#### To delete a Spot Instance data feed

- To delete a data feed, enter the following command:

```
PROMPT> ec2-delete-spot-datafeed-subscription
```

If the request is successful, the output is empty.

## API

### To delete a Spot Instance data feed

- Construct the following Query request.

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

Following is an example response. It confirms that the subscription was deleted.

```
<DeleteSpotDatafeedSubscriptionResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
 <requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId>
 <return>true</return>
</DeleteSpotDatafeedSubscriptionResponse>
```

## Programming Spot with AWS Java SDK

This section will hold the two SDK tutorials

You can AWS Java SDK to program Spot Instances:

- [Tutorial: Amazon EC2 Spot Instances \(p. 184\)](#)
- [Tutorial: Advanced Amazon EC2 Spot Request Management \(p. 194\)](#)

## Tutorial: Amazon EC2 Spot Instances

### Overview

Spot Instances allow you to bid on unused Amazon Elastic Compute Cloud (Amazon EC2) capacity and run the acquired instances for as long as your bid exceeds the current *Spot Price*. Amazon EC2 changes the Spot Price periodically based on supply and demand, and customers whose bids meet or exceed it gain access to the available Spot Instances. Like On-Demand Instances and Reserved Instances, Spot Instances provide you another option for obtaining more compute capacity.

Spot Instances can significantly lower your Amazon EC2 costs for batch processing, scientific research, image processing, video encoding, data and web crawling, financial analysis, and testing. Additionally, Spot Instances give you access to large amounts of additional capacity in situations where the need for that capacity is not urgent.

To use Spot Instances, place a Spot Instance request specifying the maximum price you are willing to pay per instance hour; this is your bid. If your bid exceeds the current Spot Price, your request is fulfilled and your instances will run until either you choose to terminate them or the Spot Price increases above your bid (whichever is sooner).

It's important to note two points:

- You will often pay less per hour than your bid. Amazon EC2 adjusts the Spot Price periodically as requests come in and available supply changes. Everyone pays the same Spot Price for that period regardless of whether their bid was higher. Therefore, you might pay less than your bid, but you will never pay more than your bid.
- If you're running Spot Instances and your bid no longer meets or exceeds the current Spot Price, your instances will be terminated. This means that you will want to make sure that your workloads and applications are flexible enough to take advantage of this opportunistic capacity.

Spot Instances perform exactly like other Amazon EC2 instances while running, and like other Amazon EC2 instances, Spot Instances can be terminated when you no longer need them. If you terminate your instance, you pay for any partial hour used (as you would for On-Demand or Reserved Instances). However, if the Spot Price goes above your bid and your instance is terminated by Amazon EC2, you will not be charged for any partial hour of usage.

This tutorial provides a quick overview of how to use the Java programming language to do the following.

- Submit a Spot Request
- Determine when the Spot Request becomes fulfilled
- Cancel the Spot Request
- Terminate associated instances

### Prerequisites

To use this tutorial you need to be signed up for Amazon Web Services (AWS). If you have not yet signed up for AWS, go to the [Amazon Web Services website](#), and click **Create an AWS Account** in the upper right corner of the page. In addition, you also need to install the [AWS Java SDK](#).

If you are using the Eclipse development environment, we recommend that you install the [AWS Toolkit for Eclipse](#). Note that the AWS Toolkit for Eclipse includes the latest version of the AWS SDK for Java.

### Step 1: Setting Up Your Credentials

To begin using this code sample, you need to populate your credentials in the `AwsCredentials.properties` file. Specifically, you need to populate your secret key and access key.

### To view your AWS access credentials

1. Go to the Amazon Web Services website at <http://aws.amazon.com>.
2. Click **My Account/Console**, and then click **Security Credentials**.
3. Under **Your Account**, click **Security Credentials**.
4. In the spaces provided, type your user name and password, and then click **Sign in using our secure server**.
5. Under **Access Credentials**, on the **Access Keys** tab, your access key ID is displayed. To view your secret key, under **Secret Access Key**, click **Show**.

Copy and paste your Access Key and Secret Access Key into the AwsCredentials.properties file. To learn more about setting up security credentials go to [Amazon EC2 Security Credentials](#).

Now that you have configured your settings, you can get started using the code in the example.

### Step 2: Setting Up a Security Group

A *security group* acts as a firewall that controls the traffic allowed in and out of a group of instances. By default, an instance is started without any security group, which means that all incoming IP traffic, on any TCP port will be denied. So, before submitting our Spot Request, we will set up a security group that allows the necessary network traffic. For the purposes of this tutorial, we will create a new security group called "GettingStarted" that allows Secure Shell (SSH) traffic from the IP address where you are running your application from. To set up a new security group, you need to include or run the following code sample that sets up the security group programmatically.

After we create an `AmazonEC2` client object, we create a `CreateSecurityGroupRequest` object with the name, "GettingStarted" and a description for the security group. Then we call the `ec2.createSecurityGroup` API to create the group.

To enable access to the group, we create an `ipPermission` object with the IP address range set to the CIDR representation of the subnet for the local computer; the "/10" suffix on the IP address indicates the subnet for the specified IP address. We also configure the `ipPermission` object with the TCP protocol and port 22 (SSH). The final step is to call `ec2.authorizeSecurityGroupIngress` with the name of our security group and the `ipPermission` object.

```
1 // Retrieves the credentials from an AwsCredentials.properties file.
2 AWS Credentials credentials = null;
3 try {
4 credentials = new PropertiesCredentials(
5 GettingStartedApp.class.getResourceAsStream("AwsCredentials.properties"));
6 } catch (IOException e1) {
7 System.out.println("Credentials were not properly entered into AwsCredentials.properties.");
8 System.out.println(e1.getMessage());
9 }
10 System.exit(-1);

// Create the AmazonEC2Client object so we can call various APIs.
11 AmazonEC2 ec2 = new AmazonEC2Client(credentials);
12
13 // Create a new security group.
14 try {
15 CreateSecurityGroupRequest securityGroupRequest = new CreateSecurity
```

```
GroupRequest("GettingStartedGroup", "Getting Started Security Group");
 ec2.createSecurityGroup(securityGroupRequest);
20 } catch (AmazonServiceException ase) {
 // Likely this means that the group is already created, so ignore.
 System.out.println(ase.getMessage());
}

25 String ipAddr = "0.0.0.0/0";

 // Get the IP of the current host, so that we can limit the Security
 // Group by default to the ip range associated with your subnet.
 try {
30 InetAddress addr = InetAddress.getLocalHost();

 // Get IP Address
 ipAddr = addr.getHostAddress() + "/10";
 } catch (UnknownHostException e) {
35 }

 // Create a range that you would like to populate.
 ArrayList<String> ipRanges = new ArrayList<String>();
 ipRanges.add(ipAddr);

40 // Open up port 22 for TCP traffic to the associated IP
 // from above (e.g. ssh traffic).
 ArrayList<IpPermission> ipPermissions = new ArrayList<IpPermission> ();
 IpPermission ipPermission = new IpPermission();
 ipPermission.setIpProtocol("tcp");
 ipPermission.setFromPort(new Integer(22));
 ipPermission.setToPort(new Integer(22));
 ipPermission.setIpRanges(ipRanges);
 ipPermissions.add(ipPermission);

50 try {
 // Authorize the ports to the used.
 AuthorizeSecurityGroupIngressRequest ingressRequest =
 new AuthorizeSecurityGroupIngressRequest("GettingStartedGroup", ip
Permissions);
 ec2.authorizeSecurityGroupIngress(ingressRequest);
 } catch (AmazonServiceException ase) {
 // Ignore because this likely means the zone has
 // already been authorized.
 System.out.println(ase.getMessage());
60 }
```

You can view this entire code sample in the `CreateSecurityGroupApp.java` code sample. Note you only need to run this application once to create a new security group.

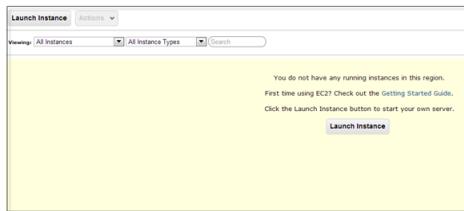
You can also create the security group using the AWS Toolkit for Eclipse. Go to the [toolkit documentation](#) for more information.

### Step 3: Submitting Your Spot Request

To submit a Spot request, you first need to determine the instance type, Amazon Machine Image (AMI), and maximum bid price you want to use. You must also include the security group we configured previously, so that you can log into the instance if desired.

There are several instance types to choose from; go to [Amazon EC2 Instance Types](#) for a complete list. For this tutorial, we will use t1.micro, the cheapest instance type available. Next, we will determine the type of AMI we would like to use. We'll use ami-8c1fce5, the most up-to-date Amazon Linux AMI available when we wrote this tutorial. The latest AMI may change over time, but you can always determine the latest version AMI by:

1. Logging into the AWS Management Console, clicking the EC2 tab, and, from the EC2 Console Dashboard, attempting to launch an instance.



2. In the window that displays AMIs, just use the AMI ID as shown in the following screen shot. Alternatively, you can use the `DescribeImages` API, but leveraging that command is outside the scope of this tutorial.



There are many ways to approach bidding for Spot instances; to get a broad overview of the various approaches you should view the [Bidding for Spot Instances](#) video. However, to get started, we'll describe three common strategies: bid to ensure cost is less than on-demand pricing; bid based on the value of the resulting computation; bid so as to acquire computing capacity as quickly as possible.

- **Reduce Cost below On-Demand** You have a batch processing job that will take a number of hours or days to run. However, you are flexible with respect to when it starts and when it completes. You want to see if you can complete it for less cost than with On-Demand Instances. You examine the Spot Price history for instance types using either the AWS Management Console or the Amazon EC2 API. For more information, go to [Viewing Spot Price History](#). After you've analyzed the price history for your desired instance type in a given Availability Zone, you have two alternative approaches for your bid:
  - You could bid at the upper end of the range of Spot Prices (which are still below the On-Demand price), anticipating that your one-time Spot request would most likely be fulfilled and run for enough consecutive compute time to complete the job.
  - Or, you could bid at the lower end of the price range, and plan to combine many instances launched over time through a persistent request. The instances would run long enough--in aggregate--to complete the job at an even lower total cost. (We will explain how to automate this task later in this tutorial.)
- **Pay No More than the Value of the Result** You have a data processing job to run. You understand the value of the job's results well enough to know how much they are worth in terms of computing costs. After you've analyzed the Spot Price history for your instance type, you choose a bid price at which the cost of the computing time is no more than the value of the job's results. You create a persistent bid and allow it to run intermittently as the Spot Price fluctuates at or below your bid.
- **Acquire Computing Capacity Quickly** You have an unanticipated, short-term need for additional capacity that is not available through On-Demand Instances. After you've analyzed the Spot Price

history for your instance type, you bid above the highest historical price to provide a high likelihood that your request will be fulfilled quickly and continue computing until it completes.

After you choose your bid price, you are ready to request a Spot Instance. For the purposes of this tutorial, we will bid the On-Demand price (\$0.03) to maximize the chances that the bid will be fulfilled. You can determine the types of available instances and the On-Demand prices for instances by going to Amazon EC2 Pricing page. To request a Spot Instance, you simply need to build your request with the parameters you chose earlier. We start by creating a `RequestSpotInstancesRequest` object. The request object requires the number of instances you want to start and the bid price. Additionally, you need to set the `LaunchSpecification` for the request, which includes the instance type, AMI ID, and security group you want to use. Once the request is populated, you call the `requestSpotInstances` method on the `AmazonEC2Client` object. The following example shows how to request a Spot Instance.

```
1 // Retrieves the credentials from a AWS Credentials.properties file.
2 AWSCredentials credentials = null;
3 try {
4 credentials = new PropertiesCredentials(
5 GettingStartedApp.class.getResourceAsStream("AwsCredentials.properties"));
6 } catch (IOException e1) {
7 System.out.println("Credentials were not properly entered into AwsCredentials.properties.");
8 System.out.println(e1.getMessage());
9 System.exit(-1);
10 }

11 // Create the AmazonEC2Client object so we can call various APIs.
12 AmazonEC2 ec2 = new AmazonEC2Client(credentials);
13
14 // Initializes a Spot Instance Request
15 RequestSpotInstancesRequest requestRequest = new RequestSpotInstancesRequest();

16 // Request 1 x t1.micro instance with a bid price of $0.03.
17 requestRequest.setSpotPrice("0.03");
18 requestRequest.setInstanceCount(Integer.valueOf(1));

19 // Setup the specifications of the launch. This includes the
20 // instance type (e.g. t1.micro) and the latest Amazon Linux
21 // AMI id available. Note, you should always use the latest
22 // Amazon Linux AMI id or another of your choosing.
23 LaunchSpecification launchSpecification = new LaunchSpecification();
24 launchSpecification.setImageId("ami-8c1fce5");
25 launchSpecification.setInstanceType("t1.micro");

26 // Add the security group to the request.
27 ArrayList<String> securityGroups = new ArrayList<String>();
28 securityGroups.add("GettingStartedGroup");
29 launchSpecification.setSecurityGroups(securityGroups);

30 // Add the launch specifications to the request.
31 requestRequest.setLaunchSpecification(launchSpecification);

32 // Call the RequestSpotInstance API.
33 RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(re
```

```
questRequest);
```

Running this code will launch a new Spot Instance Request. There are other options you can use to configure your Spot Requests. To learn more, please visit the [Java Developers: Advanced Spot Features Tutorials](#) or the [RequestSpotInstances API](#) in the Java SDK.

**Note**

You will be charged for any Spot Instances that are actually launched, so make sure that you cancel any requests and terminate any instances you launch to reduce any associated fees.

#### Step 4: Determining the State of Your Spot Request

Next, we want to create code to wait until the Spot request reaches the "active" state before proceeding to the last step. To determine the state of our Spot request, we poll the [describeSpotInstanceRequests](#) method for the state of the Spot request ID we want to monitor.

The request ID created in Step 2 is embedded in the response to our `requestSpotInstances` request. The following example code shows how to gather request IDs from the `requestSpotInstances` response and use them to populate an `ArrayList`.

```
1 // Call the RequestSpotInstance API.
2 RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(re
questRequest);
3 List<SpotInstanceRequest> requestResponses = requestResult.getSpotInstance
4 Requests();
5
6 // Setup an arraylist to collect all of the request ids we want to
7 // watch hit the running state.
8 ArrayList<String> spotInstanceRequestIds = new ArrayList<String>();
9
10 // Add all of the request ids to the hashset, so we can determine when they
11 // hit the
12 // active state.
13 for (SpotInstanceRequest requestResponse : requestResponses) {
14 System.out.println("Created Spot Request: " + requestResponse.getSpotIn
stanceRequestId());
15 spotInstanceRequestIds.add(requestResponse.getSpotInstanceRequestId());
}
```

To monitor your request ID, call the `describeSpotInstanceRequests` method to determine the state of the request. Then loop until the request is not in the "open" state. Note that we monitor for a state of not "open", rather a state of, say, "active", because the request can go straight to "closed" if there is a problem with your request arguments. The following code example provides the details of how to accomplish this task.

```
1
2 // Create a variable that will track whether there are any
3 // requests still in the open state.
4 boolean anyOpen;
5
6 do {
7 // Create the describeRequest object with all of the request ids
8 // to monitor (e.g. that we started).
```

```
 DescribeSpotInstanceRequestsRequest describeRequest = new DescribeSpot
InstanceRequestsRequest();
10 describeRequest.setSpotInstanceRequestIds(spotInstanceRequestIds);

 // Initialize the anyOpen variable to false - which assumes there
 // are no requests open unless we find one that is still open.
 anyOpen=false;
15
 try {
 // Retrieve all of the requests we want to monitor.
 DescribeSpotInstanceRequestsResult describeResult = ec2.describeS
potInstanceRequests(describeRequest);
 List<SpotInstanceRequest> describeResponses = describeResult.get
SpotInstanceRequests();
20
 // Look through each request and determine if they are all in
 // the active state.
 for (SpotInstanceRequest describeResponse : describeResponses) {
 // If the state is open, it hasn't changed since we attempted
25 // to request it. There is the potential for it to transition
 // almost immediately to closed or canceled so we compare
 // against open instead of active.
 if (describeResponse.getState().equals("open")) {
 anyOpen = true;
30 break;
 }
 }
 } catch (AmazonServiceException e) {
 // If we have an exception, ensure we don't break out of
35 // the loop. This prevents the scenario where there was
 // blip on the wire.
 anyOpen = true;
 }

40 try {
 // Sleep for 60 seconds.
 Thread.sleep(60*1000);
 } catch (Exception e) {
 // Do nothing because it woke up early.
45 }
} while (anyOpen);
```

After running this code, your Spot Instance Request will have completed or will have failed with an error that will be output to the screen. In either case, we can proceed to the next step to clean up any active requests and terminate any running instances.

### Step 5: Cleaning Up Your Spot Requests and Instances

Lastly, we need to clean up our requests and instances. It is important to both cancel any outstanding requests *and* terminate any instances. Just canceling your requests will not terminate your instances, which means that you will continue to pay for them. If you terminate your instances, your Spot requests may be canceled, but there are some scenarios—such as if you use persistent bids—where terminating your instances is not sufficient to stop your request from being re-fulfilled. Therefore, it is a best practice to both cancel any active bids and terminate any running instances.

The following code demonstrates how to cancel your requests.

```
1 try {
 // Cancel requests.
 CancelSpotInstanceRequestsRequest cancelRequest = new CancelSpotInstan
ceRequestsRequest(spotInstanceRequestIds);
5 ec2.cancelSpotInstanceRequests(cancelRequest);
} catch (AmazonServiceException e) {
 // Write out any exceptions that may have occurred.
 System.out.println("Error canceling instances");
 System.out.println("Caught Exception: " + e.getMessage());
10 System.out.println("Reponse Status Code: " + e.getStatusCode());
 System.out.println("Error Code: " + e.getErrorCode());
 System.out.println("Request ID: " + e.getRequestId());
}
```

To terminate any outstanding instances, you will need the instance ID associated with the request that started them. The following code example takes our original code for monitoring the instances and adds an `ArrayList` in which we store the instance ID associated with the `describeInstances` response.

```
1 // Create a variable that will track whether there are any requests
2 // still in the open state.
3 boolean anyOpen;
4
5 // Initialize variables.
6 ArrayList<String> instanceIds = new ArrayList<String>();
7
8 do {
9 // Create the describeRequest with all of the request ids to
10 // monitor (e.g. that we started).
11 DescribeSpotInstanceRequestsRequest describeRequest = new DescribeSpot
12 InstanceRequestsRequest();
13 describeRequest.setSpotInstanceRequestIds(spotInstanceRequestIds);
14
15 // Initialize the anyOpen variable to false, which assumes there
16 // are no requests open unless we find one that is still open.
17 anyOpen = false;
18
19 try {
20 // Retrieve all of the requests we want to monitor.
21 DescribeSpotInstanceRequestsResult describeResult = ec2.describeS
22 potInstanceRequests(describeRequest);
23 List<SpotInstanceRequest> describeResponses = describeResult.get
24 SpotInstanceRequests();
25
26 // Look through each request and determine if they are all
27 // in the active state.
28 for (SpotInstanceRequest describeResponse : describeResponses) {
29 // If the state is open, it hasn't changed since we
30 // attempted to request it. There is the potential for
31 // it to transition almost immediately to closed or
32 // canceled so we compare against open instead of active.
33 if (describeResponse.getState().equals("open")) {
34 anyOpen = true;
35 break;
36 }
37 }
38 }
39 }
40}
```

```

35 // Add the instance id to the list we will
 // eventually terminate.
 instanceIds.add(describeResponse.getInstanceId());
 }
40 } catch (AmazonServiceException e) {
 // If we have an exception, ensure we don't break out
 // of the loop. This prevents the scenario where there
 // was blip on the wire.
 anyOpen = true;
45 }

 try {
 // Sleep for 60 seconds.
 Thread.sleep(60*1000);
50 } catch (Exception e) {
 // Do nothing because it woke up early.
 }
} while (anyOpen);

```

Using the instance IDs, stored in the `ArrayList`, terminate any running instances using the following code snippet.

```

1
try {
 // Terminate instances.
 TerminateInstancesRequest terminateRequest = new TerminateInstances
Request(instanceIds);
5 ec2.terminateInstances(terminateRequest);
 } catch (AmazonServiceException e) {
 // Write out any exceptions that may have occurred.
 System.out.println("Error terminating instances");
 System.out.println("Caught Exception: " + e.getMessage());
10 System.out.println("Reponse Status Code: " + e.getStatusCode());
 System.out.println("Error Code: " + e.getErrorCode());
 System.out.println("Request ID: " + e.getRequestId());
 }

```

## Bringing It All Together

To bring this all together, we provide a more object-oriented approach that combines the preceding steps we showed: initializing the EC2 Client, submitting the Spot Request, determining when the Spot Requests are no longer in the open state, and cleaning up any lingering Spot request and associated instances. We create a class called `Requests` that performs these actions.

We also create a `GettingStartedApp` class, which has a main method where we perform the high level function calls. Specifically, we initialize the `Requests` object described previously. We submit the Spot Instance request. Then we wait for the Spot request to reach the "Active" state. Finally, we clean up the requests and instances.

The complete source code is available for download at [GitHub](#).

Congratulations! You have just completed the getting started tutorial for developing Spot Instance software with the AWS Java SDK.

## **Next Steps**

We recommend that you take the Java Developers: [Tutorial: Advanced Amazon EC2 Spot Request Management \(p. 194\)](#).

## Tutorial: Advanced Amazon EC2 Spot Request Management

### Overview

Spot Instances allow you to bid on unused Amazon Elastic Compute Cloud (Amazon EC2) capacity and run those instances for as long as your bid exceeds the current Spot Price. Amazon EC2 changes the Spot Price periodically based on supply and demand. Customers whose bids meet or exceed the Spot Price gain access to the available Spot Instances. Like On-Demand Instances and Reserved Instances, Spot Instances provide you an additional option for obtaining more compute capacity.

Spot Instances can significantly lower your Amazon EC2 costs for batch processing, scientific research, image processing, video encoding, data and web crawling, financial analysis, and testing. Additionally, Spot Instances can provide access to large amounts of additional compute capacity when your need for the capacity is not urgent.

This tutorial provides a quick overview of some advanced Spot Request features, such as detailed options to create Spot requests, alternative methods for launching Spot Instances, and methods to manage your instances. This tutorial is not meant to be a complete list of all advanced topics associated with Spot Instances. Instead, it gives you a quick reference of code samples for some of the commonly used methods for managing Spot Requests and Spot Instances.

### Prerequisites

To use this tutorial you need to be signed up for Amazon Web Services (AWS). If you have not yet signed up for AWS, go to the [Amazon Web Services website](#), and click **Create an AWS Account** in the upper right corner of the page. In addition, you also need to install the [AWS Java SDK](#).

If you are using the Eclipse development environment, we recommend that you install the [AWS Toolkit for Eclipse](#). Note that the AWS Toolkit for Eclipse includes the latest version of the AWS SDK for Java.

### Step 1: Setting Up Your Credentials

To begin using this code sample, you need to populate your credentials in the AwsCredentials.properties file. Specifically, you need to populate your secretKey and accessKey.

#### To view your AWS access credentials

1. Go to the Amazon Web Services website at <http://aws.amazon.com>.
2. Click **My Account/Console**, and then click **Security Credentials**.
3. Under **Your Account**, click **Security Credentials**.
4. In the spaces provided, type your user name and password, and then click **Sign in using our secure server**.
5. Under **Access Credentials**, on the **Access Keys** tab, your access key ID is displayed. To view your secret key, under **Secret Access Key**, click **Show**.

Copy and paste your Access Key and Secret Access Key into the AwsCredentials.properties file. To learn more about setting up security credentials go to [Amazon EC2 Security Credentials](#).

### Step 2: Setting Up a Security Group

Additionally, you need to configure your *security group*. A security group acts as a firewall that controls the traffic allowed in and out of a group of instances. By default, an instance is started without any security group, which means that all incoming IP traffic, on any TCP port will be denied. So, before submitting our Spot Request, we will set up a security group that allows the necessary network traffic. For the purposes of this tutorial, we will create a new security group called "GettingStarted" that allows Secure Shell (SSH)

traffic from the IP address where you are running your application from. To set up a new security group, you need to include or run the following code sample that sets up the security group programmatically.

After we create an `AmazonEC2` client object, we create a `CreateSecurityGroupRequest` object with the name, "GettingStarted" and a description for the security group. Then we call the `ec2.createSecurityGroup` API to create the group.

To enable access to the group, we create an `ipPermission` object with the IP address range set to the CIDR representation of the subnet for the local computer; the "/10" suffix on the IP address indicates the subnet for the specified IP address. We also configure the `ipPermission` object with the TCP protocol and port 22 (SSH). The final step is to call `ec2.authorizeSecurityGroupIngress` with the name of our security group and the `ipPermission` object.

(The following code is the same as what we used in the first tutorial.)

```
1 // Retrieves the credentials from an AWSCredentials.properties file.
AWSCredentials credentials = null;
try {
5 credentials = new PropertiesCredentials(
 GettingStartedApp.class.getResourceAsStream("AwsCredentials.properties"));
} catch (IOException e1) {
 System.out.println("Credentials were not properly entered into AwsCreden-
tials.properties.");
 System.out.println(e1.getMessage());
10 System.exit(-1);
}

// Create the AmazonEC2Client object so we can call various APIs.
AmazonEC2 ec2 = new AmazonEC2Client(credentials);
15
// Create a new security group.
try {
 CreateSecurityGroupRequest securityGroupRequest =
 new CreateSecurityGroupRequest("GettingStartedGroup", "Getting
Started Security Group");
20 ec2.createSecurityGroup(securityGroupRequest);
} catch (AmazonServiceException ase) {
 // Likely this means that the group is already created, so ignore.
 System.out.println(ase.getMessage());
}
25
String ipAddr = "0.0.0.0/0";

// Get the IP of the current host, so that we can limit the Security Group
// by default to the ip range associated with your subnet.
30 try {
 InetAddress addr = InetAddress.getLocalHost();

 // Get IP Address
 ipAddr = addr.getHostAddress() + "/10";
35 } catch (UnknownHostException e) {
}

// Create a range that you would like to populate.
ArrayList<String> ipRanges = new ArrayList<String>();
40 ipRanges.add(ipAddr);
```

```
// Open up port 22 for TCP traffic to the associated IP from
// above (e.g. ssh traffic).
ArrayList<IpPermission> ipPermissions = new ArrayList<IpPermission> ();
45 IpPermission ipPermission = new IpPermission();
ipPermission.setIpProtocol("tcp");
ipPermission.setFromPort(new Integer(22));
ipPermission.setToPort(new Integer(22));
ipPermission.setIpRanges(ipRanges);
50 ipPermissions.add(ipPermission);

try {
 // Authorize the ports to the used.
 AuthorizeSecurityGroupIngressRequest ingressRequest =
55 new AuthorizeSecurityGroupIngressRequest("GettingStartedGroup", ip
Permissions);
 ec2.authorizeSecurityGroupIngress(ingressRequest);
} catch (AmazonServiceException ase) {
 // Ignore because this likely means the zone has already
 // been authorized.
60 System.out.println(ase.getMessage());
}
```

You can view this entire code sample in the `advanced.CreateSecurityGroupApp.java` code sample. Note you only need to run this application once to create a new security group.

You can also create the security group using the AWS Toolkit for Eclipse. Go to the [toolkit documentation](#) for more information.

### Detailed Spot Instance Request Creation Options

As we explained in [Tutorial: Amazon EC2 Spot Instances \(p. 184\)](#), you need to build your request with an instance type, an Amazon Machine Image (AMI), and maximum bid price.

Let's start by creating a `RequestSpotInstanceRequest` object. The request object requires the number of instances you want and the bid price. Additionally, we need to set the `LaunchSpecification` for the request, which includes the instance type, AMI ID, and security group you want to use. After the request is populated, we call the `requestSpotInstances` method on the `AmazonEC2Client` object. An example of how to request a Spot instance follows.

(The following code is the same as what we used in the first tutorial.)

```
1
// Retrieves the credentials from an AWSCredentials.properties file.
AWSCredentials credentials = null;
try {
5 credentials = new PropertiesCredentials(
 GettingStartedApp.class.getResourceAsStream("AwsCredentials.proper
ties"));
} catch (IOException e1) {
 System.out.println("Credentials were not properly entered into AwsCre
dentials.properties.");
 System.out.println(e1.getMessage());
10 System.exit(-1);
}
```

```

 // Create the AmazonEC2Client object so we can call various APIs.
 AmazonEC2 ec2 = new AmazonEC2Client(credentials);
15
 // Initializes a Spot Instance Request
 RequestSpotInstancesRequest requestRequest = new RequestSpotInstances
Request();

 // Request 1 x t1.micro instance with a bid price of $0.03.
20 requestRequest.setSpotPrice("0.03");
 requestRequest.setInstanceCount(Integer.valueOf(1));

 // Set up the specifications of the launch. This includes the
 // instance type (e.g. t1.micro) and the latest Amazon Linux
25 // AMI id available. Note, you should always use the latest
 // Amazon Linux AMI id or another of your choosing.
 LaunchSpecification launchSpecification = new LaunchSpecification();
 launchSpecification.setImageId("ami-8c1fece5");
 launchSpecification.setInstanceType("t1.micro");
30
 // Add the security group to the request.
 ArrayList<String> securityGroups = new ArrayList<String>();
 securityGroups.add("GettingStartedGroup");
 launchSpecification.setSecurityGroups(securityGroups);
35
 // Add the launch specification.
 requestRequest.setLaunchSpecification(launchSpecification);

 // Call the RequestSpotInstance API.
40 RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(re
questRequest);

```

## Persistent vs. One-Time Requests

When building a Spot request, you can specify several optional parameters. The first is whether your request is one-time only or persistent. By default, it is a one-time request. A one-time request can be fulfilled only once, and after the requested instances are terminated, the request will be closed. A persistent request is considered for fulfillment whenever there is no Spot Instance running for the same request. To specify the type of request, you simply need to set the Type on the Spot request. This can be done with the following code.

```

1
// Retrieves the credentials from an
// AWSCredentials.properties file.
AWSCredentials credentials = null;
5 try {
 credentials = new PropertiesCredentials(
 GettingStartedApp.class.getResourceAsStream("AwsCredentials.proper
ties"));
} catch (IOException e1) {
 System.out.println("Credentials were not properly entered into AwsCre
dentials.properties.");
10 System.out.println(e1.getMessage());
 System.exit(-1);
}

// Create the AmazonEC2Client object so we can call various APIs.

```

```

15 AmazonEC2 ec2 = new AmazonEC2Client(credentials);

 // Initializes a Spot Instance Request
 RequestSpotInstancesRequest requestRequest = new RequestSpotInstances
Request();

20 // Request 1 x t1.micro instance with a bid price of $0.03.
 requestRequest.setSpotPrice("0.03");
 requestRequest.setInstanceCount(Integer.valueOf(1));

 // Set the type of the bid to persistent.
25 requestRequest.setType("persistent");

 // Set up the specifications of the launch. This includes the
 // instance type (e.g. t1.micro) and the latest Amazon Linux
 // AMI id available. Note, you should always use the latest
30 // Amazon Linux AMI id or another of your choosing.
 LaunchSpecification launchSpecification = new LaunchSpecification();
 launchSpecification.setImageId("ami-8c1fece5");
 launchSpecification.setInstanceType("t1.micro");

35 // Add the security group to the request.
 ArrayList<String> securityGroups = new ArrayList<String>();
 securityGroups.add("GettingStartedGroup");
 launchSpecification.setSecurityGroups(securityGroups);

40 // Add the launch specification.
 requestRequest.setLaunchSpecification(launchSpecification);

 // Call the RequestSpotInstance API.
 RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(re
questRequest);
45

```

### **Limiting the Duration of a Request**

You can also optionally specify the length of time that your request will remain valid. You can specify both a starting and ending time for this period. By default, a Spot request will be considered for fulfillment from the moment it is created until it is either fulfilled or canceled by you. However you can constrain the validity period if you need to. An example of how to specify this period is shown in the following code.

```

1
// Retrieves the credentials from an AWS Credentials.properties file.
AWSCredentials credentials = null;
try {
5 credentials = new PropertiesCredentials(
 GettingStartedApp.class.getResourceAsStream("AwsCredentials.proper
ties"));
} catch (IOException e1) {
 System.out.println("Credentials were not properly entered into AwsCre
dentials.properties.");
 System.out.println(e1.getMessage());
10 System.exit(-1);
}

// Create the AmazonEC2Client object so we can call various APIs.
AmazonEC2 ec2 = new AmazonEC2Client(credentials);

```

```

15 // Initializes a Spot Instance Request
16 RequestSpotInstancesRequest requestRequest = new RequestSpotInstances
17 Request();
18
19 // Request 1 x t1.micro instance with a bid price of $0.03.
20 requestRequest.setSpotPrice("0.03");
21 requestRequest.setInstanceCount(Integer.valueOf(1));
22
23 // Set the valid start time to be two minutes from now.
24 Calendar cal = Calendar.getInstance();
25 cal.add(Calendar.MINUTE, 2);
26 requestRequest.setValidFrom(cal.getTime());
27
28 // Set the valid end time to be two minutes and two hours from now.
29 cal.add(Calendar.HOUR, 2);
30 requestRequest.setValidUntil(cal.getTime());
31
32 // Set up the specifications of the launch. This includes
33 // the instance type (e.g. t1.micro)
34
35 // and the latest Amazon Linux AMI id available.
36 // Note, you should always use the latest Amazon
37 // Linux AMI id or another of your choosing.
38 LaunchSpecification launchSpecification = new LaunchSpecification();
39 launchSpecification.setImageId("ami-8c1fece5");
40 launchSpecification.setInstanceType("t1.micro");
41
42 // Add the security group to the request.
43 ArrayList<String> securityGroups = new ArrayList<String>();
44 securityGroups.add("GettingStartedGroup");
45 launchSpecification.setSecurityGroups(securityGroups);
46
47 // Add the launch specification.
48 requestRequest.setLaunchSpecification(launchSpecification);
49
50 // Call the RequestSpotInstance API.
51 RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(re
questRequest);

```

## Grouping Your Amazon EC2 Spot Instance Requests

You have the option of grouping your Spot instance requests in several different ways. We'll look at the benefits of using launch groups, Availability Zone groups, and placement groups.

If you want to ensure your Spot instances are all launched and terminated together, then you have the option to leverage a launch group. A launch group is a label that groups a set of bids together. All instances in a launch group are started and terminated together. Note, if instances in a launch group have already been fulfilled, there is no guarantee that new instances launched with the same launch group will also be fulfilled. An example of how to set a Launch Group is shown in the following code example.

```

1
// Retrieves the credentials from an AWSCredentials.properties file.
AWSCredentials credentials = null;
try {
5 credentials = new PropertiesCredentials(

```

```

 GettingStartedApp.class.getResourceAsStream("AwsCredentials.properties"));
 } catch (IOException e1) {
 System.out.println("Credentials were not properly entered into AwsCredentials.properties.");
 System.out.println(e1.getMessage());
10 System.exit(-1);
 }

 // Create the AmazonEC2Client object so we can call various APIs.
 AmazonEC2 ec2 = new AmazonEC2Client(credentials);
15
 // Initializes a Spot Instance Request
 RequestSpotInstancesRequest requestRequest = new RequestSpotInstancesRequest();

 // Request 5 x t1.micro instance with a bid price of $0.03.
20 requestRequest.setSpotPrice("0.03");
 requestRequest.setInstanceCount(Integer.valueOf(5));

 // Set the launch group.
 requestRequest.setLaunchGroup("ADVANCED-DEMO-LAUNCH-GROUP");
25
 // Set up the specifications of the launch. This includes
 // the instance type (e.g. t1.micro) and the latest Amazon Linux
 // AMI id available. Note, you should always use the latest
 // Amazon Linux AMI id or another of your choosing.
30 LaunchSpecification launchSpecification = new LaunchSpecification();
 launchSpecification.setImageId("ami-8c1fece5");
 launchSpecification.setInstanceType("t1.micro");

 // Add the security group to the request.
35 ArrayList<String> securityGroups = new ArrayList<String>();
 securityGroups.add("GettingStartedGroup");
 launchSpecification.setSecurityGroups(securityGroups);

 // Add the launch specification.
40 requestRequest.setLaunchSpecification(launchSpecification);

 // Call the RequestSpotInstance API.
 RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(re
questRequest);

```

If you want to ensure that all instances within a request are launched in the same Availability Zone, and you don't care which one, you can leverage Availability Zone groups. An Availability Zone group is a label that groups a set of instances together in the same Availability Zone. All instances that share an Availability Zone group and are fulfilled at the same time will start in the same Availability Zone. An example of how to set an Availability Zone group follows.

```

1
// Retrieves the credentials from an AWSCredentials.properties file.
AWSCredentials credentials = null;
try {
5 credentials = new PropertiesCredentials(
 GettingStartedApp.class.getResourceAsStream("AwsCredentials.properties"));

```

```

 } catch (IOException e1) {
 System.out.println("Credentials were not properly entered into AwsCredentials.properties.");
 System.out.println(e1.getMessage());
10 System.exit(-1);
 }

 // Create the AmazonEC2Client object so we can call various APIs.
 AmazonEC2 ec2 = new AmazonEC2Client(credentials);
15
 // Initializes a Spot Instance Request
 RequestSpotInstancesRequest requestRequest = new RequestSpotInstancesRequest();
 RequestSpotInstancesResult requestResult;

 // Request 5 x t1.micro instance with a bid price of $0.03.
20 requestRequest.setSpotPrice("0.03");
 requestRequest.setInstanceCount(Integer.valueOf(5));

 // Set the availability zone group.
 requestRequest.setAvailabilityZoneGroup("ADVANCED-DEMO-AZ-GROUP");
25
 // Set up the specifications of the launch. This includes the instance
 // type (e.g. t1.micro) and the latest Amazon Linux AMI id available.
 // Note, you should always use the latest Amazon Linux AMI id or another
 // of your choosing.
30 LaunchSpecification launchSpecification = new LaunchSpecification();
 launchSpecification.setImageId("ami-8c1fece5");
 launchSpecification.setInstanceType("t1.micro");

 // Add the security group to the request.
35 ArrayList<String> securityGroups = new ArrayList<String>();
 securityGroups.add("GettingStartedGroup");
 launchSpecification.setSecurityGroups(securityGroups);

 // Add the launch specification.
40 requestRequest.setLaunchSpecification(launchSpecification);

 // Call the RequestSpotInstance API.
 RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(re
questRequest);

```

You can specify an Availability Zone that you want for your Spot Instances. The following code example shows you how to set an Availability Zone.

```

1
// Retrieves the credentials from an AWS Credentials properties file.
AWSCredentials credentials = null;
try {
5 credentials = new PropertiesCredentials(
 GettingStartedApp.class.getResourceAsStream("AwsCredentials.properties"));
} catch (IOException e1) {
 System.out.println("Credentials were not properly entered into AwsCredentials.properties.");
 System.out.println(e1.getMessage());
10 System.exit(-1);
}

```

```

 }

 // Create the AmazonEC2Client object so we can call various APIs.
 AmazonEC2 ec2 = new AmazonEC2Client(credentials);
15
 // Initializes a Spot Instance Request
 RequestSpotInstancesRequest requestRequest = new RequestSpotInstances
Request();

 // Request 1 x t1.micro instance with a bid price of $0.03.
20 requestRequest.setSpotPrice("0.03");
 requestRequest.setInstanceCount(Integer.valueOf(1));

 // Set up the specifications of the launch. This includes the instance
 // type (e.g. t1.micro) and the latest Amazon Linux AMI id available.
25 // Note, you should always use the latest Amazon Linux AMI id or another
 // of your choosing.
 LaunchSpecification launchSpecification = new LaunchSpecification();
 launchSpecification.setImageId("ami-8c1fce5");
 launchSpecification.setInstanceType("t1.micro");
30
 // Add the security group to the request.
 ArrayList<String> securityGroups = new ArrayList<String>();
 securityGroups.add("GettingStartedGroup");
 launchSpecification.setSecurityGroups(securityGroups);
35
 // Set up the availability zone to use. Note we could retrieve the
 // availability zones using the ec2.describeAvailabilityZones() API. For
 // this demo we will just use us-east-1a.
 SpotPlacement placement = new SpotPlacement("us-east-1b");
40
 launchSpecification.setPlacement(placement);

 // Add the launch specification.
 requestRequest.setLaunchSpecification(launchSpecification);
45
 // Call the RequestSpotInstance API.
 RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(re
questRequest);

```

Lastly, you can specify a *placement group* if you are using High Performance Computing (HPC) Spot instances, such as cluster compute instances or cluster GPU instances. Placement groups provide you with lower latency and high-bandwidth connectivity between the instances. An example of how to set a placement group follows.

```

1
 // Retrieves the credentials from an AWSCredentials.properties file.
 AWSCredentials credentials = null;
 try {
5 credentials = new PropertiesCredentials(
 GettingStartedApp.class.getResourceAsStream("AwsCredentials.proper
ties"));
 } catch (IOException e1) {
 System.out.println("Credentials were not properly entered into AwsCre
dentials.properties.");
 System.out.println(e1.getMessage());

```

```
10 System.exit(-1);
}

// Create the AmazonEC2Client object so we can call various APIs.
AmazonEC2 ec2 = new AmazonEC2Client(credentials);
15
// Initializes a Spot Instance Request
RequestSpotInstancesRequest requestRequest = new RequestSpotInstancesRequest();

// Request 1 x t1.micro instance with a bid price of $0.03.
20 requestRequest.setSpotPrice("0.03");
requestRequest.setInstanceCount(Integer.valueOf(1));

// Set up the specifications of the launch. This includes the instance
// type (e.g. t1.micro) and the latest Amazon Linux AMI id available.
25 // Note, you should always use the latest Amazon Linux AMI id or another
// of your choosing.
LaunchSpecification launchSpecification = new LaunchSpecification();
launchSpecification.setImageId("ami-8c1fce5");
launchSpecification.setInstanceType("t1.micro");
30
// Add the security group to the request.
ArrayList<String> securityGroups = new ArrayList<String>();
securityGroups.add("GettingStartedGroup");
launchSpecification.setSecurityGroups(securityGroups);
35
// Set up the placement group to use with whatever name you desire.
// For this demo we will just use "ADVANCED-DEMO-PLACEMENT-GROUP".
SpotPlacement placement = new SpotPlacement();
placement.setGroupName("ADVANCED-DEMO-PLACEMENT-GROUP");
40 launchSpecification.setPlacement(placement);

// Add the launch specification.
requestRequest.setLaunchSpecification(launchSpecification);

45 // Call the RequestSpotInstance API.
RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(re
questRequest);
```

All of the parameters shown in this section are optional. It is also important to realize that most of these parameters—with the exception of whether your bid is one-time or persistent—can reduce the likelihood of bid fulfillment. So, it is important to leverage these options only if you need them. All of the preceding code examples are combined into one long code sample, which can be found in the `com.amazonaws.codesamples.advanced.InlineGettingStartedCodeSampleApp.java` class.

### How to Persist a Root Partition After Interruption or Termination

One of the easiest ways to manage interruption of your Spot instances is to ensure that your data is checkpointed to an Amazon Elastic Block Store (Amazon EBS) volume on a regular cadence. By checkpointing periodically, if there is an interruption you will lose only the data created since the last checkpoint (assuming no other non-idempotent actions are performed in between). To make this process easier, you can configure your Spot Request to ensure that your root partition will not be deleted on interruption or termination. We've inserted new code in the following example that shows how to enable this scenario.

In the added code, we create a `BlockDeviceMapping` object and set its associated Elastic Block Storage (EBS) to an EBS object that we've configured to not be deleted if the Spot Instance is terminated. We then add this `BlockDeviceMapping` to the `ArrayList` of mappings that we include in the launch specification.

```
1 // Retrieves the credentials from an AWS Credentials.properties file.
2 AWSCredentials credentials = null;
3 try {
4 credentials = new PropertiesCredentials(
5 GettingStartedApp.class.getResourceAsStream("AwsCredentials.properties"));
6 } catch (IOException e1) {
7 System.out.println("Credentials were not properly entered into AwsCredentials.properties.");
8 System.out.println(e1.getMessage());
9 }
10 System.exit(-1);
11 }

12 // Create the AmazonEC2Client object so we can call various APIs.
13 AmazonEC2 ec2 = new AmazonEC2Client(credentials);
14
15 // Initializes a Spot Instance Request
16 RequestSpotInstancesRequest requestRequest = new RequestSpotInstancesRequest();

17 // Request 1 x t1.micro instance with a bid price of $0.03.
18 requestRequest.setSpotPrice("0.03");
19 requestRequest.setInstanceCount(Integer.valueOf(1));

20 // Set up the specifications of the launch. This includes the instance
21 // type (e.g. t1.micro) and the latest Amazon Linux AMI id available.
22 // Note, you should always use the latest Amazon Linux AMI id or another
23 // of your choosing.
24 LaunchSpecification launchSpecification = new LaunchSpecification();
25 launchSpecification.setImageId("ami-8c1fce5");
26 launchSpecification.setInstanceType("t1.micro");
27
28 // Add the security group to the request.
29 ArrayList<String> securityGroups = new ArrayList<String>();
30 securityGroups.add("GettingStartedGroup");
31 launchSpecification.setSecurityGroups(securityGroups);
32
33 // Create the block device mapping to describe the root partition.
34 BlockDeviceMapping blockDeviceMapping = new BlockDeviceMapping();
35 blockDeviceMapping.setDeviceName("/dev/sda1");

36 // Set the delete on termination flag to false.
37 EbsBlockDevice ebs = new EbsBlockDevice();
38 ebs.setDeleteOnTermination(Boolean.FALSE);
39 blockDeviceMapping.setEbs(ebs);

40 // Add the block device mapping to the block list.
41 ArrayList<BlockDeviceMapping> blockList = new ArrayList<BlockDeviceMapping>();
42 blockList.add(blockDeviceMapping);

43 // Set the block device mapping configuration in the launch specifications.
```

```
50 launchSpecification.setBlockDeviceMappings(blockList);

 // Add the launch specification.
 requestRequest.setLaunchSpecification(launchSpecification);

55 // Call the RequestSpotInstance API.
 RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(re
questRequest);
```

Assuming you wanted to re-attach this volume to your instance on startup, you can also use the block device mapping settings. Alternatively, if you attached a non-root partition, you can specify the Amazon EBS volumes you want to attach to your Spot instance after it resumes. You do this simply by specifying a snapshot ID in your `EbsBlockDevice` and alternative device name in your `BlockDeviceMapping` objects. By leveraging block device mappings, it can be easier to bootstrap your instance.

Using the root partition to checkpoint your critical data is a great way to manage the potential for interruption of your instances. For more methods on managing the potential of interruption, please visit the [Managing Interruption](#) video.

## How to Tag Your Spot Requests and Instances

Adding [tags to EC2 resources](#) can simplify the administration of your cloud infrastructure. A form of metadata, tags can be used to create user-friendly names, enhance searchability, and improve coordination between multiple users. You can also use tags to automate scripts and portions of your processes.

To add tags to your resources, you need to tag them *after* they have been requested. Specifically, you must add a tag after a Spot request has been submitted or after the `RunInstances` call has been performed. The following code example illustrates adding tags.

```
1
/*
 * Copyright 2010-2011 Amazon.com, Inc. or its affiliates. All Rights Re
served.
 *
5 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
10 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
15 */
package com.amazonaws.codesamples.advanced;

import java.io.IOException;
import java.util.ArrayList;
20 import java.util.List;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.auth.AWS Credentials;
import com.amazonaws.auth.PropertiesCredentials;
25 import com.amazonaws.codesamples.getting_started.GettingStartedApp;
import com.amazonaws.services.ec2.AmazonEC2;
```

```
import com.amazonaws.services.ec2.AmazonEC2Client;
import com.amazonaws.services.ec2.model.CancelSpotInstanceRequestsRequest;
import com.amazonaws.services.ec2.model.CreateTagsRequest;
30 import com.amazonaws.services.ec2.model.DescribeSpotInstanceRequestsRequest;
import com.amazonaws.services.ec2.model.DescribeSpotInstanceRequestsResult;
import com.amazonaws.services.ec2.model.LaunchSpecification;
import com.amazonaws.services.ec2.model.RequestSpotInstancesRequest;
import com.amazonaws.services.ec2.model.RequestSpotInstancesResult;
35 import com.amazonaws.services.ec2.model.SpotInstanceRequest;
import com.amazonaws.services.ec2.model.Tag;
import com.amazonaws.services.ec2.model.TerminateInstancesRequest;

/**
40 * Welcome to your new AWS Java SDK based project!
*
* This class is meant as a starting point for your console-based application
that
* makes one or more calls to the AWS services supported by the Java SDK,
such as EC2,
* SimpleDB, and S3.
45 *
* In order to use the services in this sample, you need:
*
* - A valid Amazon Web Services account. You can register for AWS at:
* https://aws-portal.amazon.com/gp/aws/developer/registration/in
dex.html
50 *
* - Your account's Access Key ID and Secret Access Key:
* http://aws.amazon.com/security-credentials
*
* - A subscription to Amazon EC2. You can sign up for EC2 at:
55 * http://aws.amazon.com/ec2/
*
*/
public class InlineTaggingCodeSampleApp {
60 /**
* @param args
*/
 public static void main(String[] args) {
65 //=====
 //===== Submitting a Request =====
 //=====

 // Retrieves the credentials from an AWS Credentials.properties file.
70 AWS Credentials credentials = null;
 try {
 credentials = new PropertiesCredentials(
 GettingStartedApp.class.getResourceAsStream("AwsCredentials.properties"));
 } catch (IOException e1) {
75 System.out.println("Credentials were not properly entered into
AwsCredentials.properties.");
 System.out.println(e1.getMessage());
 System.exit(-1);
 }
 }
}
```

```
80 // Create the AmazonEC2Client object so we can
81 // call various APIs.
82 AmazonEC2 ec2 = new AmazonEC2Client(credentials);

83 // Initializes a Spot Instance Request
84 RequestSpotInstancesRequest requestRequest = new RequestSpotInstances
Request();

85 // Request 1 x t1.micro instance with a bid price of $0.03.
86 requestRequest.setSpotPrice("0.03");
87 requestRequest.setInstanceCount(Integer.valueOf(1));

88 // Set up the specifications of the launch. This includes
89 // the instance type (e.g. t1.micro) and the latest Amazon
90 // Linux AMI id available. Note, you should always use the
91 // latest Amazon Linux AMI id or another of your choosing.
92 LaunchSpecification launchSpecification = new LaunchSpecification();
93 launchSpecification.setImageId("ami-8c1fece5");
94 launchSpecification.setInstanceType("t1.micro");

95 // Add the security group to the request.
96 ArrayList<String> securityGroups = new ArrayList<String>();
97 securityGroups.add("GettingStartedGroup");
98 launchSpecification.setSecurityGroups(securityGroups);

99 // Add the launch specifications to the request.
100 requestRequest.setLaunchSpecification(launchSpecification);

101 //=====
102 //===== Getting the Request ID from the Request =====//
103 //=====

104 // Call the RequestSpotInstance API.
105 RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(re
questRequest);
106 List<SpotInstanceRequest> requestResponses = requestResult.getSpotIn
stanceRequests();

107 // Set up an arraylist to collect all of the request ids we want to
108 // watch hit the running state.
109 ArrayList<String> spotInstanceRequestIds = new ArrayList<String>();

110 // Add all of the request ids to the hashset, so we can
111 // determine when they hit the active state.
112 for (SpotInstanceRequest requestResponse : requestResponses) {
113 System.out.println("Created Spot Request: "+requestResponse.getSpot
InstanceId());
114 spotInstanceRequestIds.add(requestResponse.getSpotInstanceRequestId());
115 }

116 //=====
117 //===== Tag the Spot Requests =====//
118 //=====

119 // Create the list of tags we want to create
120 ArrayList<Tag> requestTags = new ArrayList<Tag>();
121 requestTags.add(new Tag("keyname1","value1"));

122 //=====
123 //===== Tag the Spot Requests =====//
124 //=====
```

```
// Create a tag request for the requests.
135 CreateTagsRequest createTagsRequest_requests = new CreateTagsRequest();
createTagsRequest_requests.setResources(spotInstanceRequestIds);
createTagsRequest_requests.setTags(requestTags);

// Try to tag the Spot request submitted.
140 try {
 ec2.createTags(createTagsRequest_requests);
} catch (AmazonServiceException e) {
 // Write out any exceptions that may have occurred.
 System.out.println("Error terminating instances");
 System.out.println("Caught Exception: " + e.getMessage());
 System.out.println("Reponse Status Code: " + e.getStatusCode());
 System.out.println("Error Code: " + e.getErrorCode());
 System.out.println("Request ID: " + e.getRequestId());
}

150 //=====
//===== Determining the State of the Spot Request =====//
//=====

155 // Create a variable that will track whether there are any
// requests still in the open state.
boolean anyOpen;

160 // Initialize variables.
ArrayList<String> instanceIds = new ArrayList<String>();

do {
 // Create the describeRequest with tall of the request
 // id to monitor (e.g. that we started).
165 DescribeSpotInstanceRequestsRequest describeRequest = new DescribeSpotInstanceRequestsRequest();
 describeRequest.setSpotInstanceRequestIds(spotInstanceRequestIds);

 // Initialize the anyOpen variable to false - which assumes there are
no requests open unless
 // we find one that is still open.
170 anyOpen = false;

 try {
 // Retrieve all of the requests we want to monitor.
 DescribeSpotInstanceRequestsResult describeResult = ec2.describeSpotInstanceRequests(describeRequest);
175 List<SpotInstanceRequest> describeResponses = describeResult.getSpotInstanceRequests();

 // Look through each request and determine if they are all
 // in the active state.
 for (SpotInstanceRequest describeResponse : describeResponses) {
 // If the state is open, it hasn't changed since we
 // attempted to request it. There is the potential
 // for it to transition almost immediately to closed or
 // canceled so we compare against open instead of active.
 if (describeResponse.getState().equals("open")) {
 anyOpen = true;
 break;
 }
 }
 }
}
```

```

190 // Add the instance id to the list we will
191 // eventually terminate.
192 instanceIds.add(describeResponse.getInstanceId());
193 }
194 } catch (AmazonServiceException e) {
195 // If we have an exception, ensure we don't break out
196 // of the loop. This prevents the scenario where there
197 // was blip on the wire.
198 anyOpen = true;
199 }

200 try {
201 // Sleep for 60 seconds.
202 Thread.sleep(60*1000);
203 } catch (Exception e) {
204 // Do nothing because it woke up early.
205 }
206 } while (anyOpen);

//=====
//===== Tag the Spot Instances ======
//=====

210

// Create the list of tags we want to create
ArrayList<Tag> instanceTags = new ArrayList<Tag>();
instanceTags.add(new Tag("keyname1","value1"));

215
// Create a tag request for instances.
CreateTagsRequest createTagsRequest_instances = new CreateTagsRequest();
createTagsRequest_instances.setResources(instanceIds);
createTagsRequest_instances.setTags(instanceTags);

220
// Try to tag the Spot instance started.
try {
221 ec2.createTags(createTagsRequest_instances);
222} catch (AmazonServiceException e) {
223 // Write out any exceptions that may have occurred.
224 System.out.println("Error terminating instances");
225 System.out.println("Caught Exception: " + e.getMessage());
226 System.out.println("Reponse Status Code: " + e.getStatusCode());
227 System.out.println("Error Code: " + e.getErrorCode());
228 System.out.println("Request ID: " + e.getRequestId());
229}

//=====
//===== Canceling the Request ======
//=====

230

235

try {
236 // Cancel requests.
237 CancelSpotInstanceRequestsRequest cancelRequest = new CancelSpotIn
238stanceRequestsRequest(spotInstanceRequestIds);
239 ec2.cancelSpotInstanceRequests(cancelRequest);
240} catch (AmazonServiceException e) {
241 // Write out any exceptions that may have occurred.
242 System.out.println("Error canceling instances");
243 System.out.println("Caught Exception: " + e.getMessage());

```

```
245 System.out.println("Reponse Status Code: " + e.getStatusCode());
246 System.out.println("Error Code: " + e.getErrorCode());
247 System.out.println("Request ID: " + e.getRequestId());
248 }
249
250 //=====
251 //===== Terminating any Instances =====//
252 //=====
253 try {
254 // Terminate instances.
255 TerminateInstancesRequest terminateRequest = new TerminateInstancesRequest(instanceIds);
256 ec2.terminateInstances(terminateRequest);
257 } catch (AmazonServiceException e) {
258 // Write out any exceptions that may have occurred.
259 System.out.println("Error terminating instances");
260 System.out.println("Caught Exception: " + e.getMessage());
261 System.out.println("Reponse Status Code: " + e.getStatusCode());
262 System.out.println("Error Code: " + e.getErrorCode());
263 System.out.println("Request ID: " + e.getRequestId());
264 }
265 } // main
}
```

Tags are a simple first step toward making it easier to manage your own cluster of instances. To read more about tagging Amazon EC2 resources, go to [Using Tags](#) in the Amazon Elastic Compute Cloud User Guide.

### Bringing It All Together

To bring this all together, we provide a more object-oriented approach that combines the steps we showed in this tutorial into one easy to use class. We instantiate a class called `Requests` that performs these actions. We also create a `GettingStartedApp` class, which has a main method where we perform the high level function calls.

The complete source code is available for download at [GitHub](#).

Congratulations! You've completed the Advanced Request Features tutorial for developing Spot Instance software with the AWS SDK for Java.

## Starting Clusters on Spot Instances

Grids are a form of distributed computing that enable a user to leverage multiple instances to perform parallel computations. Customers—such as [Numerate](#), [Scribd](#), and the [University of Barcelona/University of Melbourne](#)—use Grid Computing with Spot Instances because this type of architecture can take advantage of Spot Instance's built-in elasticity and low prices to get work done faster at a more cost-effective price.

To get started, a user will break down the work into discrete units called jobs, and then submit that work to a “master node.” These jobs will be queued up, and a process called a “scheduler” will distribute that work out to other instances in the grid, called “worker nodes.” After the result is computed by the worker node, the master node is notified, and the worker node can take the next operation from the queue. If the job fails or the instance is interrupted, the job will automatically be re-queued by the scheduler process.

As you work to architect your application, it is important to choose the appropriate amount of work to be included in your job. We recommend breaking your jobs down into a logical grouping based on the time it would take to process. Typically, you will want to create a workload size less than an hour, so that if

you have to process the workload again, it doesn't cost you additional money (you don't pay for the hour if we interrupt your instance).

Many customers use a Grid scheduler, such as Oracle Grid Engine or UniCloud, to set up a cluster. If you have long-running workloads, the best practice is to run the master node on On-Demand or Reserved Instances, and run the worker nodes on Spot or a mixture of On-Demand, Reserved, and Spot Instances. Alternatively, if you have a workload that is less than an hour or you are running a test environment, you may want to run all of your instances on Spot. No matter the setup, we recommend that you create a script to automatically re-add instances that may be interrupted. Some existing tools—StarCluster, for example—can help you manage this process.

## **Quick Look: How to Launch a Cluster on Spot Video**

Chris Dagdigian, from AWS Solution Provider [BioTeam](#), provides a quick overview of how to start a cluster from scratch in about 10 to 15 minutes on Amazon EC2 Spot Instances using StarCluster. StarCluster is an open source tool created by a lab at MIT that makes it easy to set up a new Oracle Grid Engine cluster. In this video, Chris walks through the process of installing, setting up, and running simple jobs on a cluster. Chris also leverages Spot Instances, so that you can potentially get work done faster and potentially save between 50 percent to 66 percent. [How to Launch a Cluster on Spot](#)

# Reserved Instances

Amazon Elastic Compute Cloud (Amazon EC2) Reserved Instances is a pricing model that enables you to reserve capacity for your EC2 instances and lowers your average instance cost. With Reserved Instances, you pay a low, one-time fee for the capacity reservation and then receive a significant discount on the hourly charge for your instances. When you want to use your reserved capacity, you launch an EC2 instance with the same configuration as the reserved capacity that you purchased. Amazon Web Services (AWS) will automatically apply the discounted hourly rate that is associated with your capacity reservation. You are charged the discounted hourly rate for your EC2 instance for as long as you own the Reserved Instance. When the term of your Reserved Instance ends, you can continue using the EC2 instance without interruption. However, you will now be charged at the On-Demand rate.

Reserved Instances can provide substantial savings over owning your own hardware or running only On-Demand instances, as well as help assure that the capacity you need is available to you when you require it.

To purchase an Amazon EC2 Reserved Instance, you must select an instance type (such as m1.small), platform (Linux/UNIX, Windows, or Windows with SQL Server), location (region and Availability Zone), and term (either one year or three years). If you want your Reserved Instance to run on a specific Linux/UNIX platform, you must identify that platform when you purchase the reserved capacity. Then, when you're ready to use the Reserved Instance that you purchased, you must choose an Amazon Machine Image (AMI) that runs that specific Linux/UNIX platform, along with any other specifications you identified during the purchase.

For example, if you require a one-year, *SUSE Linux*, m1.medium Reserved Instance in the Singapore region, purchasing a one-year, *Linux/UNIX*, m1.medium Reserved Instance in the Singapore region will not give you the capacity guarantee and pricing benefit. Both your Reserved Instance purchase and the instance you launch must specify the same *SUSE Linux* product platform.

## Reserved Instance Overview

The following information will help you get started working with Amazon EC2 Reserved Instances:

- Complete all the prerequisite tasks first—such as registration, signing up, and installing the tools—so you can start working with Reserved Instances. For more information, see [Getting Started with Reserved Instances \(p. 213\)](#).
- Before you buy and sell Reserved Instances, you can learn more about them by reading [Steps for Using Reserved Instances \(p. 214\)](#).
- Standard one- and three-year terms for Reserved Instances are available for purchase from AWS, and non-standard terms are available for purchase from third-party resellers through the Reserved Instance Marketplace.
- Optimize your Reserved Instance costs by selecting the pricing model that best matches how often you plan to use your instances. For more information, see [Choosing Reserved Instances Based on Your Usage Plans \(p. 217\)](#).
- Learn more about the pricing benefit of Reserved Instances. For more information, see [Understanding the Pricing Benefit of Reserved Instances \(p. 241\)](#).
- Understand Reserved Instance pricing tiers and how to take advantage of discount pricing. For more information, see [Understanding Discount Pricing Tiers \(p. 229\)](#).
- You can sell your unused Reserved Instances in the Reserved Instance Marketplace. The Reserved Instance Marketplace makes it possible for sellers who have Reserved Instances that they no longer need to find buyers who are looking to purchase additional capacity. Reserved Instances bought and sold through the Reserved Instance Marketplace work like any other Reserved Instances. For more information, see [Reserved Instance Marketplace \(p. 243\)](#).

For a checklist that summarizes requirements for working with Reserved Instances and the Reserved Instance Marketplace, see [Requirements Checklist for Reserved Instances \(p. 272\)](#).

## What Do You Want to Do Next?

- Learn:
  - [Getting Started with Reserved Instances \(p. 213\)](#)
  - [Buying Reserved Instances \(p. 219\)](#)
  - [Selling in the Reserved Instance Marketplace \(p. 245\)](#)
  - [Using Reserved Instances in Amazon VPC \(p. 215\)](#)
- Start:
  - [Becoming a Buyer \(p. 220\)](#)
  - [Purchasing Reserved Instances \(p. 221\)](#)
  - [Obtaining Information About Your Reserved Instances \(p. 237\)](#)
  - [Registering as a Seller \(p. 246\)](#)
  - [Listing Your Reserved Instance \(p. 251\)](#)

## Getting Started with Reserved Instances

### Topics

- [Get Set Up \(p. 213\)](#)
- [Steps for Using Reserved Instances \(p. 214\)](#)
- [Using Reserved Instances in Amazon VPC \(p. 215\)](#)
- [Tools for Working with Reserved Instances \(p. 215\)](#)

You can use Amazon Elastic Compute Cloud (Amazon EC2) Reserved Instances to reserve capacity for your instances and get the benefits of lower-cost computing. With Reserved Instances you pay a low, one-time fee and in turn receive a significant discount on the hourly charge for your instance. Reserved Instances can provide substantial savings over owning your own hardware or running only On-Demand instances, as well as help assure that the capacity you need is available to you when you require it. This topic takes you through the fundamentals you need to get started with Reserved Instances.

## Get Set Up

Before you get started working with Reserved Instances, you should complete the following tasks:

- Sign up.

To work with Reserved Instances, read and complete the instructions described in [Getting Started with Amazon EC2 Linux Instances \(p. 8\)](#), which provides information on signing up for your Amazon EC2 account and credentials.

- Install the tools.

You can use the Amazon EC2 tools—AWS Management Console, Amazon EC2 Command Line Interface (CLI) tools, or the Amazon EC2 API—to work with EC2 Reserved Instances and search for offerings. For more information, see [Tools for Working with Reserved Instances \(p. 215\)](#).

If you want to start working with Reserved Instances using specific tools, see [AWS Management Console \(p. 215\)](#), the [Command Line Interface Tools \(p. 216\)](#), or the [API \(p. 217\)](#).

## Steps for Using Reserved Instances

There are four sets of steps to follow when you use Reserved Instances. You can *purchase* Amazon EC2 Reserved Instances, and then you can *launch* them. You can *view* the Reserved Instances you have, and you can *sell* unused Reserved Instances in the Reserved Instance Marketplace. (Restrictions apply. For information, see [Requirements Checklist for Reserved Instances \(p. 272\)](#).) This section describes purchasing, launching, viewing, and selling Reserved Instances.

You can use the AWS Management Console, the Amazon EC2 CLI tools, or the Amazon EC2 API to perform any of these tasks. Before you get started, you need to set up the prerequisite accounts and tools. For more information, see [Get Set Up \(p. 213\)](#).

1. Purchase.

- a. Determine how much capacity you want to reserve. Specify the following criteria for your instance reservation.
  - Platform (for example, Linux/UNIX).

**Note**

When you want your Reserved Instance to run on a specific Linux/UNIX platform, you must identify the specific platform when you purchase the reserved capacity. Then, when you launch your instance with the intention of using the reserved capacity you purchased, you must choose the Amazon Machine Image (AMI) that runs that specific Linux/UNIX platform, along with any other specifications you identified during the purchase.

- Instance type (for example, m1.small).
  - Term (time period) over which you want to reserve capacity (one or three years).
  - Tenancy specification, if you want to reserve capacity for your instance to run in single-tenant hardware (*dedicated* tenancy, as opposed to *shared*).
  - Region and Availability Zone where you want to run the instance.
- b. Choose the offering type that best addresses how much you want to pay and how often you plan to run your instances.
    - Heavy Utilization
    - Medium Utilization
    - Light Utilization

For more information about these offering types, see [Choosing Reserved Instances Based on Your Usage Plans \(p. 217\)](#).

- c. Search for offerings that meet the criteria you specified.
- d. Purchase offerings that fulfill your requirements.

For more information, see [Purchasing Reserved Instances \(p. 221\)](#).

2. Launch.

To use your Reserved Instance, launch an On-Demand EC2 instance with the same criteria as your Reserved Instance (the region, Availability Zone, instance type, and platform specified when you purchased your Reserved Instance). See step 1.

**Note**

Reserved Instance pricing benefits and capacity guarantees automatically apply to any running EC2 instances you have that aren't already covered by a reservation.

For more information, see [Launching EC2 Instances \(p. 274\)](#).

3. View.

You can view the Reserved Instances that you own or that are available to your account, and confirm that your instances are running as specified.

For more information, see [Obtaining Information About Your Reserved Instances \(p. 237\)](#).

4. Sell Reserved Instance capacity that you no longer need.
  - a. Register as a seller in the Reserved Instance Marketplace using the Seller Registration wizard. For more information, see [Registering as a Seller \(p. 246\)](#).

**Note**

Not every customer can sell in the Reserved Instance Marketplace and not all Reserved Instances can be sold in the Reserved Instance Marketplace. For information, see [Requirements Checklist for Reserved Instances \(p. 272\)](#).

- b. Decide on a price for the Reserved Instances that you want to sell. For more information, see [Pricing Your Reserved Instances \(p. 251\)](#).
- c. List your Reserved Instances. For more information, see [Listing Your Reserved Instance \(p. 251\)](#).
- d. Find out how you get paid when your Reserved Instances are sold. For more information, see [Getting Paid \(p. 271\)](#).

## Using Reserved Instances in Amazon VPC

To launch Reserved Instances in Amazon Virtual Private Cloud (Amazon VPC), you must either have an account that supports a default VPC or you must purchase an Amazon VPC Reserved Instance.

If your account does not support a default VPC, you must purchase an Amazon VPC Reserved Instance by selecting a platform that includes *Amazon VPC* in its name. For more information, see [Detecting Your Supported Platforms and Whether You Have a Default VPC](#). For information about Amazon VPC, see [What is Amazon VPC?](#) in the *Amazon Virtual Private Cloud User Guide*.

If your account supports a default VPC, the list of platforms available does not include *Amazon VPC* in its name because all platforms have default subnets. In this case, if you launch an instance with the same configuration as the capacity you reserved and paid for, that instance is launched in your default VPC and the capacity guarantees and billing benefits are applied to your instance. For information about default VPCs, see [Your Default VPC and Subnets](#) in the *Amazon Virtual Private Cloud User Guide*.

You can also choose to purchase Reserved Instances that are physically isolated at the host hardware level by specifying *dedicated* as the instance tenancy. For more information about Dedicated Instances, see [Using EC2 Dedicated Instances Within Your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

## Tools for Working with Reserved Instances

You can use the [AWS Management Console \(p. 215\)](#), the [Command Line Interface Tools \(p. 216\)](#), or the [API \(p. 217\)](#) to list or search for available Amazon EC2 Reserved Instances, purchase reserved capacity, manage your Reserved Instance, and sell your unused Reserved Instances.

If you use the CLI tools, first you should read and complete the instructions described in [Setting Up the Amazon EC2 Command Line Interface Tools on Linux/UNIX \(p. 560\)](#). The Getting Started topic walks you through setting up your environment for use with the CLI tools.

### AWS Management Console

The AWS Management Console has tools specifically designed for Reserved Instances tasks. You can find them in the Amazon EC2 console. You will also find general tools that you can use to manage the instances launched when you use your Reserved Instances.

- The **Reserved Instances** page is where you work with your Reserved Instances.

Purchase Reserved Instances									
Viewing: All Reserved Instances		Purchase More Like This		Sell Reserved Instances					
RI ID	Instance Type	Zone	Term	Offering Type	Fixed Price	Usage Price	Instance Count	Recurring Charges	State
4f4a0fc7-0d34-4ae9-a... 11 micro	us-east-1d	12-16 months	Heavy Utilization	\$56.00	\$ 0.00	1	-	Hourly: \$0.005	● active
b1bae23-03b4-4a0b-b... 11 micro	us-east-1a	12 months	Light Utilization	\$23.00	\$ 0.012	1	-	Hourly: \$0.005	● active
d10f7af1-042a-452a-b... 11 micro	us-east-1a	12-16 months	Heavy Utilization	\$56.00	\$ 0.00	1	-	Hourly: \$0.005	● active
d10f7af1-7c04-422e-8... m1 large	us-east-1a	12 months	Light Utilization	\$276.00	\$ 0.156	1	-	Hourly: \$0.095	● active
9386e4-43a-f1d2-4... c1 medium	us-east-1c	12 months	Heavy Utilization	\$24.00	\$ 0.00	1	-	Hourly: \$0.095	● active
9386e4-43a-f1d2-4... c1 medium	us-east-1c	12 months	Light Utilization	\$24.00	\$ 0.120	1	-	Hourly: \$0.095	● active
f127bd27-ec4-4f29-b... t1 micro	us-east-1c	12 months	Heavy Utilization	\$62.00	\$ 0.00	1	-	Hourly: \$0.005	● active
f127bd27-ec4-4f29-b... t1 micro	us-east-1c	12 months	Light Utilization	\$23.00	\$ 0.012	1	-	Hourly: \$0.005	● active

- Use the **Purchase Reserved Instances** page to specify the details of the Reserved Instances you want to purchase.

**Purchase Reserved Instances**

Platform:	Linux/UNIX	Term:	Any
Instance Type:	t1 micro	Tenancy:	Default
Availability Zone:	Any	Offering Type:	Any

To find a Reserved Instance offering, complete the following steps:  
 1. Specify the offering details and click **Search**.  
 2. Select the Reserved Instances and specify the quantity that you want, and **Add** them to your cart.  
 3. Click **View Cart** to view your cart before purchasing the Reserved Instances.  
 4. Click **Purchase** to complete your purchase.

Note: Additional taxes may apply.

Reserved Instances sold through the Reserved Instance Marketplace are identical to those sold by Amazon Web Services, except they may have different prices and terms. For more information about the Reserved Instance Marketplace, go to the [Reserved Instance Marketplace](#) web page.

**Cancel**      You currently have no items in your cart.      **View Cart**

- Use the **Instances** page to launch your Reserved Instances and manage them.

Launch Instance						
Viewing: All Instances		All Instance Types		Actions		
<input type="checkbox"/>	Name	Instance	AMI ID	Type	State	Lifecycle
<input type="checkbox"/>	empty	i-544bba7	ami-3275ee5b	m1.small	● running	normal
<input type="checkbox"/>	empty	i-43dbc321	ami-3275ee5b	m1.small	● running	normal
<input type="checkbox"/>	empty	i-41dbc323	ami-3275ee5b	m1.small	● running	normal
<input type="checkbox"/>	empty	i-45dbc327	ami-3275ee5b	m1.small	● running	normal
<input type="checkbox"/>	empty	i-4bdbc329	ami-3275ee5b	m1.small	● running	normal
<input type="checkbox"/>	empty	i-49dbc32b	ami-3275ee5b	m1.small	● running	normal
<input type="checkbox"/>	empty	i-4fdbc32d	ami-3275ee5b	m1.small	● running	normal
<input type="checkbox"/>	empty	i-4ddbc32f	ami-3275ee5b	m1.small	● running	normal
<input type="checkbox"/>	empty	i-53dbc331	ami-3275ee5b	m1.small	● running	normal
<input type="checkbox"/>	empty	i-57dbc335	ami-3275ee5b	m1.small	● running	normal

## Command Line Interface Tools

To purchase Reserved Instances, or sell them in the Reserved Instance Marketplace, you can use Amazon EC2 command line interface (CLI) tools specifically designed for these tasks. To manage the instances when your Reserved Instances are launched, use the same commands in the CLI tools that you use for any other Amazon EC2 instances.

The following table lists the commands in the CLI tools that you use specifically for Reserved Instances tasks.

Task	CLI
List Reserved Instances that you have purchased.	<a href="#">ec2-describe-reserved-instances</a>
View the Reserved Instances offerings that are available for purchase.	<a href="#">ec2-describe-reserved-instances-offerings</a>
Create a listing of the Reserved Instances you want to sell in the Reserved Instance Marketplace.	<a href="#">ec2-create-reserved-instances-listing</a>

Task	CLI
View the details of your Reserved Instance listings in the Reserved Instance Marketplace.	<a href="#">ec2-describe-reserved-instances-listings</a>
Purchase a Reserved Instance.	<a href="#">ec2-purchase-reserved-instances-offering</a>
Cancel your active Reserved Instances listing in the Reserved Instance Marketplace.	<a href="#">ec2-cancel-reserved-instances-listing</a>

For information about CLI commands, see the [Amazon Elastic Compute Cloud Command Line Reference](#).

## API

To purchase Reserved Instances, you use API calls specifically designed for these tasks. To manage the instances when your Reserved Instances are launched, use the same API calls that you use for any other Amazon EC2 instances.

The following table lists the API calls you use for Reserved Instances tasks.

Task	API
List Reserved Instances that you have purchased.	<a href="#">DescribeReservedInstances</a>
View the Reserved Instances offerings that are available for purchase.	<a href="#">DescribeReservedInstancesOfferings</a>
Create a listing of the Reserved Instances you want to sell in the Reserved Instance Marketplace.	<a href="#">CreateReservedInstancesListing</a>
View the details of your Reserved Instance listings in the Reserved Instance Marketplace.	<a href="#">DescribeReservedInstancesListings</a>
Purchase a Reserved Instance.	<a href="#">PurchaseReservedInstancesOffering</a>
Cancel your active Reserved Instances listing in the Reserved Instance Marketplace.	<a href="#">CancelReservedInstancesListing</a>

For information about API actions, see the [Amazon Elastic Compute Cloud API Reference](#).

## Choosing Reserved Instances Based on Your Usage Plans

You can select a Reserved Instance fee structure based on how often you plan to use your instance. We offer three Reserved Instance types to address your projected utilization of the instance: *Heavy Utilization*, *Medium Utilization*, or *Light Utilization*. (To use these Reserved Instance types, make sure you have API version 2011-11-01 or later.)

*Heavy Utilization* Reserved Instances can be used to enable workloads that have a consistent baseline of capacity, or they can run steady-state workloads. Heavy Utilization Reserved Instances require the highest upfront commitment. However, if you plan to run your Reserved Instances around 35 percent of the time or more for a three-year term, you may be able to earn the largest savings (up to around 65 percent off of the On-Demand price if your instance utilization is 100 percent) of any of the offering types. Unlike other Reserved Instances offering types, with Heavy Utilization Reserved Instances, you pay a one-time fee, followed by a lower hourly fee for the duration of the term *regardless* of whether or not your instance is running.

*Medium Utilization* Reserved Instances are the best option if you plan to use your Reserved Instances a substantial amount of the time, but want either a lower one-time fee or the flexibility to stop paying for your instance when you shut it off. Medium Utilization is a cost-effective option when you plan to run your Reserved Instances approximately between 19 percent and 35 percent of the time over the Reserved Instance term. This option can save you up to 59 percent off of the On-Demand price. With Medium Utilization Reserved Instances, you pay a slightly higher one-time fee than with Light Utilization Reserved Instances, but you receive lower hourly usage rates when you run an instance. (This offering type is equivalent to the Reserved Instance offering available before API version 2011-11-01.)

*Light Utilization* Reserved Instances are ideal for periodic workloads that run only a couple of hours a day or a few days per week. Some use cases, such as disaster recovery, also require reserved capacity to meet potential demand without notice. Using Light Utilization Reserved Instances, you pay a one-time fee followed by a discounted hourly usage fee when your instance is running. You start saving when your instance is running approximately between 11 percent and 19 percent of the time over the Reserved Instance term, and you can save up to 49 percent off of the On-Demand rates over the entire term of your Reserved Instance.

**Note**

With *Light Utilization* and *Medium Utilization* Reserved Instances, you pay a one-time upfront fee and then only pay the hourly price when you use the instance. With *Heavy Utilization* Reserved Instances, you pay a one-time upfront fee and commit to paying an hourly rate for every hour of the Reserved Instance's term *whether or not you use it*.

Remember that discounted usage fees for Reserved Instance purchases are tied to your specifications of type and Availability Zone of your instance. If you shut down a running EC2 instance on which you have been getting a discounted rate as a result of a Reserved Instance purchase, and the term of the Reserved Instance has not yet expired, you will continue to get the discounted rate if you launch another instance with the same specifications during the remainder of the term.

The following table summarizes the differences between the Reserved Instances offering types.

### Reserved Instance Offerings

Offering	Upfront Cost	Usage Fee	Advantage
Heavy Utilization	Highest	Lowest hourly fee. Applied to the whole term whether or not you're using the Reserved Instance.	Lowest overall cost if you plan to utilize your Reserved Instances more than approximately 35 percent of the time over a 3-year term.
Medium Utilization	Average	Hourly usage fee charged for each hour you use the instance. We encourage you to turn off your instances when you aren't using them so you won't be charged for them.	Suitable for elastic workloads or when you expect moderate usage, approximately between 19 percent and 35 percent of the time over a 3-year term.

Offering	Upfront Cost	Usage Fee	Advantage
Light Utilization	Lowest	Hourly usage fee charged. Highest fees of all the offering types, but they apply only when you're using the Reserved Instance. We strongly encourage you to turn off your instances when you aren't using them so you won't be charged for them.	Highest overall cost if you plan to run all of the time; however it's the lowest overall cost if you anticipate you will use your Reserved Instances infrequently, approximately between 11 percent and 19 percent of the time over a 3-year term.

## Buying Reserved Instances

You can purchase Amazon EC2 Reserved Instances with one- or three-year terms from Amazon Web Services (AWS) or you can purchase EC2 Reserved Instances from third-party sellers who own EC2 Reserved Instances that they no longer need. Reserved Instances bought from third parties and sold through the Reserved Instance Marketplace work like Reserved Instances purchased from AWS, and the purchase process is the same. The only differences are that Reserved Instances purchased from third parties will have less than a full term remaining, and they can be sold at different upfront prices.

For a buyer, the Reserved Instance Marketplace provides increased selection and flexibility by allowing you to search for Reserved Instances that most closely match your preferred combination of instance type, region, and duration.

It is important to note that once you have purchased a Reserved Instance (either from a third-party seller in the Reserved Instance Marketplace or from AWS), you cannot cancel your purchase. However, you can always sell the Reserved Instance if your needs change. For information about selling your Reserved Instance in the Reserved Instance Marketplace, see [Selling in the Reserved Instance Marketplace \(p. 245\)](#).

This buyer's guide contains the following sections:

- [Buyer Overview \(p. 244\)](#)—Provides an overview of what you need to get started buying in the Reserved Instance Marketplace.
- [Becoming a Buyer \(p. 220\)](#)—Discusses what you need to do to become a buyer in the Reserved Instance Marketplace, the information you have to disclose, and the reasons why certain information is necessary.
- [Purchasing Reserved Instances \(p. 221\)](#)—Walks you through the purchase process, which involves tasks that you likely will be repeating if you decide to use the Reserved Instance Marketplace.
- [Reading Your Statement \(Invoice\) \(p. 227\)](#)—Helps you understand your bill.

For general information about the Reserved Instance Marketplace, see [Reserved Instance Marketplace \(p. 243\)](#). For information about selling Reserved Instances in the Reserved Instance Marketplace, see [Selling in the Reserved Instance Marketplace \(p. 245\)](#). For basic information about Reserved Instances, see [Reserved Instances \(p. 212\)](#).

## How Buying in the Marketplace Works

Before you start using the Reserved Instance Marketplace, you must first create an account with AWS. To make a purchase in the Reserved Instance Marketplace, you specify the details of the Reserved Instances that you want to purchase in the AWS Management Console search wizard. You then are presented with options that match your request. Any Reserved Instance you select must be purchased just as it is listed—that is, you cannot change the term, the instance type, and so on.

After you select Reserved Instances to purchase, AWS will provide you a quote on the total cost of your selections. When you decide to proceed with the purchase, AWS will automatically place a *limit price* on the purchase price, so the total cost of your Reserved Instances will not exceed the amount you were quoted. If the price rises for any reason, AWS will automatically return you to the previous screen and let you know that your purchase did not complete because the price had changed. In addition, if at the time of purchase, there are offerings similar to your choice but at a lower price, AWS will sell you the offerings at the lower price instead of your higher-priced choice.

The Reserved Instance pricing tier discounts only apply to purchases made from AWS. These discounts do not apply to purchases of third-party Reserved Instances. For example, if you purchase \$250,000 worth of Reserved Instances from a third party in the Reserved Instance Marketplace, and then you purchase another set of third-party Reserved Instances that puts your total list price above the first discount pricing tier threshold, you will not get the 10 percent discount associated with that tier. However, if after your purchases from the marketplace cross a discount pricing threshold, you then purchase a Reserved Instance from AWS, your purchase from AWS will be discounted. For more information, see [Understanding Discount Pricing Tiers \(p. 229\)](#).

You can determine if your Reserved Instance transaction has completed by looking at the **Reserved Instances** page in the Amazon EC2 console, or the results of the `ec2-describe-reserved-instances` command, or the `DescribeReservedInstances` action. If the purchase is successful, your Reserved Instance will transition from the *pending-payment* state to the *active* state.

When buying in the marketplace, if your payment fails, the console, the `ec2-describe-reserved-instances` command, and the `DescribeReservedInstances` action will display this failed transaction by showing the Reserved Instance that you attempted to purchase to be *payment-failed*, changing from the previous *payment-pending* state.

## Becoming a Buyer

Becoming a buyer is simple and easy. If you already have an Amazon Web Services (AWS) account, you are ready to start purchasing. For more information, see the following sections:

- [Understanding the Information a Buyer Discloses \(p. 220\)](#)
- [Purchasing Reserved Instances \(p. 221\)](#)

If you don't have an AWS account, you first have to sign up and create an account with AWS. For more information, see [Getting Started with Amazon EC2 Linux Instances \(p. 8\)](#). If you are new to Reserved Instances, see [Getting Started with Reserved Instances \(p. 213\)](#).

### Understanding the Information a Buyer Discloses

Some basic information about the buyer will be shared with the seller. If you are the buyer, your ZIP code and country information will be provided to the seller in the disbursement report. This information will enable sellers to calculate any necessary transaction taxes that they have to remit to the government (such as sales tax or value-added tax). In rare circumstances, AWS might have to provide the seller with your email address, so that the seller can contact you regarding questions related to the sale (for example, tax questions).

For similar reasons, AWS will share the legal entity name of the seller on the buyer's purchase invoice. In addition, if you need additional information about the seller for tax or related reasons, you can call AWS Customer Service.

### Next Steps

After you have signed up with AWS, you can begin buying Reserved Instances in the Reserved Instance Marketplace.

- To find the Reserved Instances that address your specific business needs, see [Purchasing Reserved Instances \(p. 221\)](#).
- To understand your invoice, see [Reading Your Statement \(Invoice\) \(p. 227\)](#).
- To sell your unused Reserved Instances, see [Selling in the Reserved Instance Marketplace \(p. 245\)](#).

For information about Reserved Instances, see [Reserved Instances \(p. 212\)](#).

## Purchasing Reserved Instances

The procedure for buying Amazon EC2 Reserved Instances from third parties in the Reserved Instance Marketplace is essentially the same as the procedure for purchasing Reserved Instances from Amazon Web Services (AWS). You can purchase Reserved Instances in the Reserved Instance Marketplace using the AWS Management Console, the EC2 command line interface (CLI) tools, or the EC2 API.

### AWS Management Console

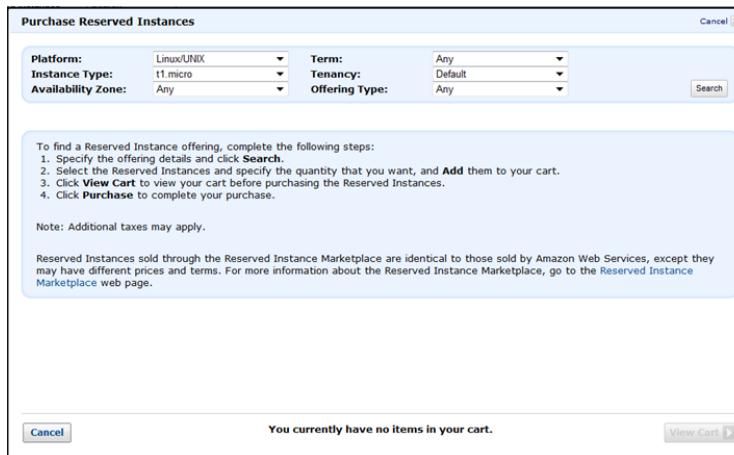
#### To find and purchase a Reserved Instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Reserved Instances** in the **Navigation** pane.
3. In the **Reserved Instances** page, click **Purchase Reserved Instances**.



Purchase Reserved Instances								
Purchase More Like These								
Sell Reserved Instances								
<a href="#">Viewing: All Reserved Instances</a> <a href="#">Search</a>								
RI ID	Instance Type	Zone	Term	Offering Type	Fixed Price	Usage Price	Instance Cost	Recurring Charges
<input type="checkbox"/> 1ba8x2e3-5fed-4e80-b1	t1.micro	us-east-1a	12 months	Light Utilization	\$23.0	\$0.012	1	-
<input type="checkbox"/> d197a91-692b-4162-b5	t1.micro	us-east-1a	12 16 months	Heavy Utilization	\$55.80	\$0.00	1	Hourly: \$0.005
<input type="checkbox"/> d197a91-7cc0-452a-85	m1.large	us-east-1a	12 months	Light Utilization	\$276.0	\$0.156	1	-

4. In the **Purchase Reserved Instances** page, specify the details of the Reserved Instances you want to purchase (use familiar filters like **Platform**, **Instance Type**, **Availability Zone**, **Term**, and **Tenancy**), and click **Search**.



**Purchase Reserved Instances**

Platform:	Linux/UNIX	Term:	Any
Instance Type:	t1.micro	Tenancy:	Default
Availability Zone:	Any	Offering Type:	Any

To find a Reserved Instance offering, complete the following steps:  
 1. Specify the offering details and click **Search**.  
 2. Select the Reserved Instances and specify the quantity that you want, and **Add** them to your cart.  
 3. Click **View Cart** to view your cart before purchasing the Reserved Instances.  
 4. Click **Purchase** to complete your purchase.

Note: Additional taxes may apply.

Reserved Instances sold through the Reserved Instance Marketplace are identical to those sold by Amazon Web Services, except they may have different prices and terms. For more information about the Reserved Instance Marketplace, go to the [Reserved Instance Marketplace](#) web page.

[Cancel](#) [View Cart](#)

You currently have no items in your cart.

The **Purchase Reserved Instances** wizard will display a list of Reserved Instances that meet your search criteria.

5. Select the Reserved Instances that you want, enter the quantity that you want to purchase, and click **Add to Cart**. You can continue to select more Reserved Instances and add them to your cart.

The **Seller** column indicates whether the seller is a **3rd Party** seller or **AWS**. Notice that the **Term** column gives you non-standard terms if the seller is a third-party seller. At the bottom of the page, the **Purchase Reserved Instances** wizard keeps a running tally of the total in your cart.

Seller	Term	Effective Rate	Upfront Price	Hourly Rate	Availability Zone	Offering Type	Quantity Available	Desired Quantity	Add to Cart
AWS	36 months	\$0.043	\$514.00	\$0.023	us-west-2c	Heavy Utilization	Unlimited	1	
AWS	36 months	\$0.043	\$514.00	\$0.023	us-west-2a	Heavy Utilization	Unlimited	1	
AWS	36 months	\$0.043	\$514.00	\$0.023	us-west-2b	Heavy Utilization	Unlimited	1	
3rd Party	31 months	\$0.048	\$498.91	\$0.026	us-west-2b	Heavy Utilization	1	1	
AWS	36 months	\$0.049	\$430.00	\$0.033	us-west-2b	Medium Utilization	Unlimited	1	
AWS	36 months	\$0.049	\$430.00	\$0.033	us-west-2a	Medium Utilization	Unlimited	1	
AWS	36 months	\$0.049	\$430.00	\$0.033	us-west-2c	Medium Utilization	Unlimited	1	
3rd Party	29 months	\$0.057	\$402.52	\$0.038	us-west-2b	Medium Utilization	1	1	
AWS	36 months	\$0.061	\$192.00	\$0.054	us-west-2c	Light Utilization	Unlimited	1	
AWS	36 months	\$0.061	\$192.00	\$0.054	us-west-2b	Light Utilization	Unlimited	1	
AWS	36 months	\$0.061	\$192.00	\$0.054	us-west-2a	Light Utilization	Unlimited	1	

Your cart: 1 Reserved Instance, Total Due Now: \$257.00  
 Additional taxes may apply.

- Click **View Cart** to see a summary of the Reserved Instances that you have selected.

If you want to add more Reserved Instances to your cart, click **Add More To Cart**. If you want to remove an item from your cart, click **Delete**.

Or, click **Cancel** if you want to start over or search for a different set of Reserved Instances.

Shopping Cart			
Items to buy now			
	Upfront Price	Hourly Price	Quantity
Heavy Utilization Linux/UNIX m1.small in us-west-2c for 36 months Payment terms: Upfront + Every Hour Sold By: Amazon <a href="#">Delete</a>	\$257.00	\$0.012	1
Light Utilization Linux/UNIX m1.medium in us-west-2a for 12 months Payment terms: Upfront + Hours Used Sold By: Amazon <a href="#">Delete</a>	\$122.00	\$0.068	1

Your cart: 2 Reserved Instances, Total Due Now: \$379.00  
 Additional taxes may apply.

- Click **Purchase** when you have all the Reserved Instances you want to purchase, and you want to check out.

#### Note

If at the time of purchase, there are offerings similar to your choice but with a lower price, AWS will sell you the offerings at the lower price.

Your purchase is complete.

8. To verify your order, go to the **Reserved Instances** page in the EC2 console.

The **Reserved Instances** page displays a list of Reserved Instances that belong to your account, including the new Reserved Instance that you just purchased.

You can use your Reserved Instance any time after your purchase is complete. This means that the **State** of your Reserved Instance has changed from *payment-pending* to *active*. To use your Reserved Instance, you launch an instance in the same way you launch an On-Demand instance. Just make sure to specify the same criteria that you specified for your Reserved Instance. AWS will automatically charge you the lower hourly rate. You do not have to restart your instance.

## Amazon EC2 CLI

### To find and purchase a Reserved Instance

1. Use `ec2-describe-reserved-instances-offerings` to get a list of Reserved Instance offerings that match your specifications. In this example, we'll check to see what m1.small, Linux/UNIX Reserved Instances are available in the sa-east-1b Availability Zone.

```
PROMPT> ec2-describe-reserved-instances-offerings -t m1.small -z sa-east-1b
-d Linux/UNIX --headers
```

Amazon EC2 returns output similar to the following example:

```
PROMPT> ec2-describe-reserved-instances-offerings
Type Source ReservedInstancesOfferingId AvailabilityZone InstanceType Duration
 FixedPrice UsagePrice ProductDescription Currency InstanceTenancy Offering
Type
OFFERING AWS 4b2293b4-3236-49f5-978d-a74c3example sa-east-1b m1.small 3y
574.0 0.0 Linux/UNIX USD default Heavy Utilization
Type Frequency Amount
RECURRING-CHARGE Hourly 0.021
OFFERING AWS 3a98bf7d-07e1-4b33-8e11-e5314example sa-east-1b m1.small 3y
473.0 0.031 Linux/UNIX USD default Medium Utilization
OFFERING AWS 438012d3-5fc5-4e49-a88e-273edexample sa-east-1b m1.small 3y
203.0 0.055 Linux/UNIX USD default Light Utilization
OFFERING AWS d586503b-bb92-41fa-9065-e5b90example sa-east-1b m1.small 1y
372.94 0.0 Linux/UNIX USD default Heavy Utilization
Type Frequency Amount
RECURRING-CHARGE Hourly 0.03
OFFERING AWS ceb6a579-b235-41e2-9aad-15a23example sa-east-1b m1.small 1y
307.13 0.04 Linux/UNIX USD default Medium Utilization
OFFERING AWS 649fd0c8-4ffb-443d-824d-eae3fexample sa-east-1b m1.small 1y
131.63 0.07 Linux/UNIX USD default Light Utilization
OFFERING 3rd Party b6121943-9faf-4350-8047-bc6d4example sa-east-1b m1.small
 10m - 0.032 Linux/UNIX USD default Medium Utilization
Type Count Price
PRICING_DETAIL 2 $1.2
OFFERING 3rd Party 08edcff2-8143-4c1d-b23c-e4c1lexample sa-east-1b m1.small
 5m - 0.032 Linux/UNIX USD default Medium Utilization
Type Count Price
PRICING_DETAIL 19 $1.2
PRICING_DETAIL 4 $1.23
```

The preceding output shows a part of the overall offerings that are available.

**Tip**

You can filter this list to return only certain types of Reserved Instances offerings of interest to you. For more information about how to filter the results, see [ec2-describe-reserved-instances-offerings](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

2. From the list of available Reserved Instances, purchase the Reserved Instances that meet your requirements. To purchase a Reserved Instance, use the following command.

```
PROMPT> ec2-purchase-reserved-instances-offering --offering offering --in
stance-count count
```

Amazon EC2 returns output similar to the following:

```
PURCHASE af9f760e-c1c1-449b-8128-1342dexample 438012d3-80c7-42c6-9396-
a209cexample
```

The response includes the offering ID and a reservation ID.

3. Write down and save the reservation ID for future reference.
4. Verify the purchase.

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

Amazon EC2 returns output similar to the following:

RESERVEDINSTANCE	af9f760e-c1c1-449b-8128-1342dexample	sa-east-1b
m1.small	1y	227.5
	0.03	Linux/UNIX
		Active

You can run your Reserved Instance any time after your purchase is complete. To run your Reserved Instance, you launch it in the same way you launch an On-Demand instance. Make sure to specify the same criteria that you specified for your Reserved Instance. AWS will automatically charge you the lower hourly rate.

## Amazon EC2 API

### To find and purchase a Reserved Instance

1. Use `DescribeReservedInstancesOfferings` to get a list of Reserved Instance offerings that match your specifications. In this example, we'll check to see the available Linux/UNIX, Heavy Utilization Reserved Instances.

```
https://ec2.amazonaws.com/?Action=DescribeReservedInstancesOfferings
&MaxResults=50
&ProductDescription=Linux%2FUNIX
&OfferingType=Heavy+Utilization
&AUTHPARAMS
```

#### Note

When using the Query API the “/” is denoted as “%2F”.

Following is an example response.

```
<DescribeReservedInstancesOfferingsResponse xmlns='http://ec2.amazonaws.com/doc/2012-08-15/'>
 <requestId>768e52ac-20f5-42b1-8559-e70e9example</requestId>
 <reservedInstancesOfferingsSet>
 <item>
 <reservedInstancesOfferingId>d0280f9e-afc1-47f3-9899-c3a2cexample</reservedInstancesOfferingId>
 <instanceType>m1.xlarge</instanceType>
 <availabilityZone>us-east-1a</availabilityZone>
 <duration>25920000</duration>
 <fixedPrice>195.0</fixedPrice>
 <usagePrice>0.0</usagePrice>
 <productDescription>Linux/UNIX</productDescription>
 <instanceTenancy>dedicated</instanceTenancy>
 <currencyCode>USD</currencyCode>
 <offeringType>Heavy Utilization</offeringType>
 <recurringCharges>
 <item>
 <frequency>Hourly</frequency>
 <amount>0.2</amount>
 </item>
 </recurringCharges>
 <marketplace>true</marketplace>
 <pricingDetailsSet>
 <item>
```

```
<price>195.0</price>
<count>1</count>
</item>
<item>
<price>310.0</price>
<count>1</count>
</item>
<item>
<price>377.0</price>
<count>1</count>
</item>
<item>
<price>380.0</price>
<count>1</count>
</item>
</pricingDetailsSet>
</item>
<item>
<reservedInstancesOfferingId>649fd0c8-7846-46b8-8f84-a6400example</reservedInstancesOfferingId>
<instanceType>m1.large</instanceType>
<availabilityZone>us-east-1a</availabilityZone>
<duration>94608000</duration>
<fixedPrice>1200.0</fixedPrice>
<usagePrice>0.0</usagePrice>
<productDescription>Linux/UNIX</productDescription>
<instanceTenancy>default</instanceTenancy>
<currencyCode>USD</currencyCode>
<offeringType>Heavy Utilization</offeringType>
<recurringCharges>
<item>
<frequency>Hourly</frequency>
<amount>0.052</amount>
</item>
</recurringCharges>
<marketplace>false</marketplace>
<pricingDetailsSet/>
</item>
</reservedInstancesOfferingsSet>
<nextToken>QUUVo/0S3X6nEBjSQZR/pRRLCPP/5Lrx79Wyxexample</nextToken>
</DescribeReservedInstancesOfferingsResponse>
```

2. From the list of available Reserved Instances in the previous example, select the marketplace offering and specify a limit price.

```
https://ec2.amazonaws.com/?Action=PurchaseReservedInstancesOffering
&ReservedInstancesOfferingId=d0280f9e-afcl-47f3-9899-c3a2cexample
&InstanceCount=1
&LimitPrice.Amount=200
&AUTHPARAMS
```

Following is an example response.

```
<PurchaseReservedInstancesOfferingResponse xmlns="http://ec2.amazonaws.com/doc/2012-08-15/">
<requestId>59dbff89-35bd-4eac-99ed-be587example</requestId>
```

```
<reservedInstancesId>e5a2ff3b-7d14-494f-90af-0b5d0example</reservedInstancesId>
</PurchaseReservedInstancesOfferingResponse>
```

3. To verify the purchase, check for your new Reserved Instance.

```
http://ec2.amazonaws.com/?Action=DescribeReservedInstances
&AUTHPARAMS
```

Following is an example response:

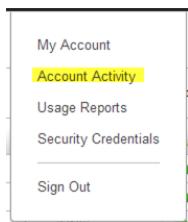
```
<DescribeReservedInstancesResponse xmlns='http://ec2.amazonaws.com/doc/2012-08-15/'>
 <requestId>ebe3410a-8f37-441d-ae11-2e78eexample</requestId>
 <reservedInstancesSet>
 <item>
 <reservedInstancesId>e5a2ff3b-7d14-494f-90af-0b5d0example</reservedInstancesId>
 <instanceType>m1.xlarge</instanceType>
 <availabilityZone>us-east-1a</availabilityZone>
 <start>2012-08-23T15:19:31.071Z</start>
 <duration>25920000</duration>
 <fixedPrice>195.0</fixedPrice>
 <usagePrice>0.0</usagePrice>
 <instanceCount>1</instanceCount>
 <productDescription>Linux/UNIX</productDescription>
 <state>active</state>
 <instanceTenancy>dedicated</instanceTenancy>
 <currencyCode>USD</currencyCode>
 <offeringType>Heavy Utilization</offeringType>
 <recurringCharges>
 <item>
 <frequency>Hourly</frequency>
 <amount>0.2</amount>
 </item>
 </recurringCharges>
 </item>
 </reservedInstancesSet>
</DescribeReservedInstancesResponse>
```

You can run your Reserved Instance any time after your purchase is complete. To run your Reserved Instance, you launch it in the same way you launch an On-Demand EC2 instance. Make sure to specify the same criteria that you specified for your Reserved Instance. AWS will automatically charge you the lower hourly rate.

## Reading Your Statement (Invoice)

### Account Activity

You can find out about the charges and fees to your account by viewing your **Account Activity** page in the AWS Management Console. You can access it by clicking the drop-down arrow beside your account name.



The **Account Activity** page will show all charges against your account—such as upfront and one-time fees and recurring charges. You can get both a summary of all your charges and a detailed list of your charges.

The upfront charges from your purchase of third-party Reserved Instances in the Reserved Instance Marketplace will be listed in the **AWS Marketplace Charges** section, with the name of the seller displayed beside it. However, all recurring or usage charges for these Reserved Instances will be listed in the **AWS Service Charges** section.

Account Activity      Welcome | Sign In | Sign Out  
 Account Number: 12345678901234567890

You are eligible for the AWS Free Usage Tier. See the Getting Started Guide AWS Free Usage Tier to learn how to get started with the free usage tier.

Monitor your estimated charges. Enable Now to begin setting billing alerts that automatically e-mail you when charges reach a threshold you define. [Learn More](#)

**This Month's Activity as of August 29, 2012**  
The statement period for this report is August 1 - August 31, 2012. The charges on this page currently show activity through approximately 08/29/2012 02:01 GMT.

Summary	
<b>AWS Service Charges</b> New Purchases and Adjustments ( <a href="#">More Info</a> ) <input type="checkbox"/> View charges and download PDF	\$69.60
<b>AWS Marketplace Charges</b> New Purchases and Adjustments ( <a href="#">More Info</a> ) <input type="checkbox"/> View charges and download PDF	\$7.91
<b>Total new charges for this statement</b> <input type="checkbox"/> Payment Summary <b>Outstanding balance for this statement</b>	\$77.51 -\$77.51 <b>\$0.00</b>

Details	
<a href="#">Expand All Services</a>   <a href="#">Collapse All Services</a> <a href="#">Printer Friendly Version</a>	
<b>AWS Service Charges</b> <input type="checkbox"/> Amazon Elastic Compute Cloud (\$69.60) <small>Download Usage Report &gt;</small> <input type="checkbox"/> Amazon Simple Notification Service (\$0.00) <small>Download Usage Report &gt;</small> <input type="checkbox"/> VAT to be collected (\$0.00)	
<b>AWS Marketplace Charges</b> <input type="checkbox"/> Amazon Elastic Compute Cloud (\$1.00) (sold by thoughtbot) <input type="checkbox"/> RI Marketplace: m1.small Linux/UNIX RI in sa-east-1a (\$6.91) (sold by thoughtbot) <input type="checkbox"/> VAT to be collected (\$0.00)	

\* Usage and recurring charges for this statement period will be charged on your next billing date, September 1, 2012. Estimated charges shown on this page, or shown on any notifications that we send to you, may differ from your actual charges for this statement period. This is because estimated charges presented on this page do not include usage charges accrued during this statement period after the date you view this page. Similarly, information about estimated charges sent to you in a notification do not include usage charges accrued during this statement period after the date we send you the notification. One-time fees and subscription charges are assessed separately from usage and recurring charges, on the date that they occur.

You can view the charges online, and you can also download a PDF rendering of the charge information.

Summary			
AWS Service Charges			\$69.60
<b>New Purchases and Adjustments</b> <small>(More Info)</small>			\$69.60
<input checked="" type="checkbox"/>	<a href="#">View charges and download PDFs</a>		
 AWS Services: Subscription charge		August 24, 2012	\$1.20
 AWS Services: Subscription charge		August 24, 2012	\$2.40
 AWS Services: Subscription charge		August 22, 2012	\$6.00
 AWS Services: Subscription charge		August 22, 2012	\$24.00
 AWS Services: Subscription charge			
 AWS Services: Subscription charge			
 AWS Services: Subscription charge			
 AWS Services: Subscription charge			
 AWS Services: Subscription charge			
<b>Total New Purchases and Adjustment</b>			
AWS Marketplace Charges			
New Purchases and Adjustments			<small>(More Info)</small>
<input checked="" type="checkbox"/>	<a href="#">View charges and download PDFs</a>		
 AWS Marketplace: Subscription charge			
 AWS Marketplace: Subscription charge			
 AWS Marketplace: Subscription charge			
 AWS Marketplace: Subscription charge			
 AWS Marketplace: Subscription charge			
 AWS Marketplace: Subscription charge			
 AWS Marketplace: Subscription charge			
<b>Total New Purchases and Adjustment</b>			
Total new charges for this statement			
 <b>Amazon Web Services Statement</b> <small>Email or talk to us about your AWS account or bill, visit <a href="http://aws.amazon.com/contact-us/">aws.amazon.com/contact-us/</a></small> <hr/> <b>Statement Summary</b> <small>Statement Number: Statement Date: August 20, 2012 \$1.20</small> <hr/> <b>TOTAL AMOUNT DUE ON August 20, 2012 \$1.20</b>			
<p>This statement is for the billing period August 1 - August 31, 2012</p> <p>Greetings from Amazon Web Services, we're writing to provide you with an electronic statement of your transactions on the AWS Marketplace. Additional information regarding your bill, individual service charge details, and your account history are available on the Account Activity Page.</p>			
Summary			
<b>AWS Marketplace Charges</b>			\$1.20
 1 x Amazon Elastic Compute Cloud (one time fee)			\$1.20
Charges			\$0.00
Credits			\$0.00
Tax			\$0.00
<b>Total for this statement</b>			\$1.20
<small>• Details of services from Japan on which consumption tax is included are provided on the Account Activity Page, visit <a href="http://aws.amazon.com/">aws.amazon.com/</a></small> <small>• This statement is not an invoice. It is a statement of transactions on the AWS Marketplace. If you need an invoice, please contact the seller listed in the description of the transaction.</small>			
Detail			
Amazon Elastic Compute Cloud sold by ThoughtPot			\$1.20
 1 x Amazon Elastic Compute Cloud (one time fee)			\$1.20
Availability Zone - GRU2			
m1.small			
Charges			\$0.00
Estimated US sales tax to be collected			\$0.00

The **Detail** section contains information about the Reserved Instance—such as the Availability Zone, instance type, cost, and number of instances. It also includes the name of the seller of the Reserved Instances that you purchased.

# Understanding Discount Pricing Tiers

You can use Amazon EC2 Reserved Instance pricing tiers to reduce your computing costs as your business grows. With Reserved Instance pricing tiers, you are entitled to a discount on your upfront fees and your usage fees when the total list price of your Reserved Instance upfront fees per region goes over the discount threshold. *List price* is the amount AWS charges at the undiscounted tier level at the time of purchase. List price is different from paid price. *Paid price* is the actual amount that an account pays for the Reserved Instances. If the Reserved Instance was purchased with a discount or the account qualifies for a discount, the paid price will be lower than the list price.

## Note

Paid price is the same as fixed price when you work with Reserved Instance tools in the AWS Management Console, command line interface (CLI) tools, and the API. The *fixed price* value—displayed in the AWS Management Console, the CLI tools, and the API—shows the price you paid per instance for all active Reserved Instances you purchased.

This section discusses key concepts you need to understand to take advantage of the discount pricing that is built into the Reserved Instance pricing tiers.

- Reserved Instance Pricing Tiers (p. 230)
  - How Reserved Instance Pricing Tiers Work (p. 230)
  - Current Limitations (p. 236)

## Reserved Instance Pricing Tiers

The following table shows the discount that is applied when the total price of the upfront fees for active Reserved Instances per region reaches the threshold of each pricing tier.

Tier level	Total price of upfront fees for active Reserved Instances per region	Discount applied to upfront and usage fees
Tier 0	\$0 - \$250,000	Standard Reserved Instance upfront and usage fees
Tier 1	\$250,000 - \$2,000,000	10 percent discount
Tier 2	\$2,000,000 - \$5,000,000	20 percent discount
Tier 3	Over \$5,000,000	<a href="#">Contact Us</a>

## How Reserved Instance Pricing Tiers Work

You can receive discounts on your purchases of future Reserved Instances based on the total list price of the upfront fees for active Reserved Instances you have for each Amazon Web Services (AWS) region. When the total list price goes above certain price points (that is, discount tier level thresholds), any future purchase of Reserved Instances will be charged at a discounted rate (as long as your total list price stays above that price point). If a single purchase takes you over the threshold of a discount tier, then the portion of that purchase that is above the price threshold will be charged at the discounted rate. Refer to the table in the previous section for tiers and discount price points.

### Note

Amazon EC2 Reserved Instance purchases are the only purchases that will apply toward your Amazon EC2 Reserved Instance pricing tiers. And the Amazon EC2 Reserved Instance pricing tiers apply only to Amazon EC2 Reserved Instance purchases.

Here's an example: Let's assume you currently have \$200,000 worth of active Reserved Instances in the us-east-1 region. You purchase 75 Reserved Instances at the list price of \$1000 each. That's a total of \$75,000, which brings the total amount you have paid for active Reserved Instances to \$275,000. Since the discount pricing threshold is \$250,000, the first \$50,000 of your new purchase would not receive a discount. The remaining \$25,000, which exceeds the discount pricing threshold, would be discounted by 10 percent (\$2,500). This means you will only be charged \$22,500 for the remainder of your purchase (25 instances), and you will be charged discounted usage rates for those 25 Reserved Instances.

If your account is part of a consolidated billing account, you can benefit from the Reserved Instance pricing tiers. A consolidated billing account aggregates into a single list price the list prices of all of the active Reserved Instances accounts in a region that are part of the consolidated billing account. When the total list price of active Reserved Instances for the consolidated billing accounts reaches the discounted tier level, any Reserved Instances purchased after this point by a member of the consolidated account will be charged at the discounted rate (as long as the total list price for that consolidated account stays above that price point).

Here's how Reserved Instance purchases work with consolidated billing: Let's assume that two accounts—A and B—are part of a consolidated billing account. All the active Reserved Instances in the consolidated billing account are in one region. Account A has Reserved Instances worth \$135,000; Account B has Reserved Instances worth \$115,000. The total upfront cost of the consolidated bill of accounts A and B is \$250,000. Remember, \$250,000 is the discount pricing threshold. This means that when either or both of the A and B accounts purchase additional Reserved Instances, the cost of the new purchases will be discounted by 10 percent. So, when account B purchases Reserved Instances at a list price of \$15,000, the consolidated account will only be charged \$13,500 for the new Reserved Instances (\$15,000 minus

the 10 percent discount of \$1500 equals \$13,500), and account B will be charged discounted usage rates for those new Reserved Instances.

For more information about how the benefits of Reserved Instances apply to consolidated billing accounts, see [Reserved Instances and Consolidated Billing \(p. 242\)](#).

## Determining Your Pricing Tier Level

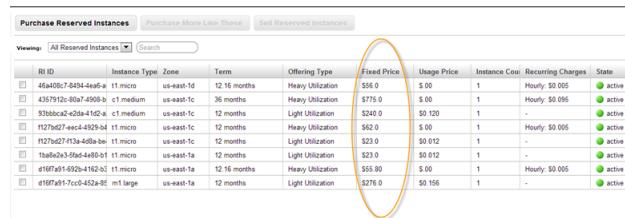
You can learn which Reserved Instance pricing tier your account is in by determining the total list price of the purchases of your account per region. *List price* is the amount charged at the undiscounted tier at the time of purchase. List price is different from the paid price—the actual amount that an account paid—for the Reserved Instances. Paid price is the fixed price value—displayed in the AWS Management Console, the CLI, and the API—per instance for all active Reserved Instances you purchased. List price is typically the same as or higher than the fixed price of a Reserved Instance. So, to determine the Reserved Instance pricing tier for your account, get the total for all the fixed price in a region. Typically, you can use the total value as basis for determining your tier level.

Alternatively, you can determine your tier level by looking at your bills and using the spreadsheet to add up all the list prices.

The following list shows how the *fixed price* values are returned when you use the console, the CLI, and the API:

- AWS Management Console

In the AWS Management Console, use the **Reserved Instances** page of the [Amazon EC2 console](#) to find out the pricing tier in which your account belongs. The **Fixed Price** column shows the amount you paid for each of your active Reserved Instances.



RI ID	Instance Type	Zone	Term	Offering Type	Fixed Price	Usage Price	Instance Count	Recurring Charges	State
46a405c7-0494-4ea9-a... t1.micro	us-east-1d		12 months	Heavy Utilization	\$56.0	\$0.00	1	Hourly: \$0.005	● active
43f7912c-80a7-400b-a... c1 medium	us-east-1c		36 months	Heavy Utilization	\$775.0	\$0.00	1	Hourly: \$0.095	● active
93bbcc42-e204-412d-a... c1 medium	us-east-1c		12 months	Light Utilization	\$240.0	\$0.120	1	-	● active
f127bd27-ee44-4929-b... t1.micro	us-east-1c		12 months	Heavy Utilization	\$62.0	\$0.00	1	Hourly: \$0.005	● active
f127bd27-f13a-4d0b-a... t1.micro	us-east-1c		12 months	Light Utilization	\$23.0	\$0.012	1	-	● active
1bae9e3-59bd-4a80-b... t1.micro	us-east-1a		12 months	Light Utilization	\$23.0	\$0.012	1	-	● active
d10f7af1-9024-4162-a... us-east-1a	us-east-1a		12 months	Heavy Utilization	\$55.80	\$0.00	1	Hourly: \$0.005	● active
d10f7af1-7c0-452a-8... ml large	us-east-1a		12 months	Light Utilization	\$276.0	\$0.156	1	-	● active

- Amazon EC2 Command Line Interface (CLI)

Use the `ec2-describe-reserved-instances` command to find out the pricing tier in which your account belongs. The `ec2-describe-reserved-instances` command will return `FixedPrice` information showing the amount you paid for all your active Reserved Instances.

Your `ec2-describe-reserved-instances` command should look like the following example:

```
PROMPT> ec2-describe-reserved-instances --headers
```

Amazon EC2 returns output similar to the following example:

```
PROMPT> ec2-describe-reserved-instances
Type ReservedInstancesId AvailabilityZone InstanceType ProductDescription
Duration FixedPrice UsagePrice InstanceCount Start State Currency InstanceTen
ancy OfferingType
RESERVEDINSTANCES f127bd27-f0e9-43bb-89f5-1b8c030bc8f9 us-east-1b m1.small
Linux/UNIX 1y 227.5 0.03 1 2011-11-28T16:17:12+0000 default active USD Medium
Utilization
RESERVEDINSTANCES f127bd27-f62e-4763-9e09-ee24f8cccf5d us-east-1b m1.large
```

```
Windows with SQL Server 3y 3400.0 0.35 1 2011-12-02T06:13:25+0000 default
active USD Medium Utilization
RESERVEDINSTANCES 1ba8e2e3-e8b2-4637-a124-ec9c11495ac9 us-east-1d m1.small
Linux/UNIX (Amazon VPC) 1y 280.0 0.035 1 2011-12-02T06:05:28+0000 dedicated
active USD Medium Utilization
RESERVEDINSTANCES 46a408c7-89fe-4b02-bb5e-ecb9bbc510eb us-east-1d m2.xlarge
Linux/UNIX 1y 1000.0 0.5 1 2011-12-02T06:05:27+0000 default active USD Lower
Utilization
RESERVEDINSTANCES af9f760e-96ae-4d12-8abe-8eb1e7a2bbdb us-east-1d t1.micro
Linux/UNIX 1y 54.0 0.0070 10 2011-12-02T06:12:09+0000 default active USD Me
dium Utilization
RESERVEDINSTANCES 46a408c7-ae07-4611-9d1c-f6d3a947f8d3 us-east-1a m1.small
Linux/UNIX (Amazon VPC) 1y 227.5 0.03 1 2011-11-08T18:00:02+0000 default
active USD Medium Utilization
RESERVEDINSTANCES bbbc9749-1211-4134-a7e7-0cdfe1caca5 us-east-1a t1.micro
Linux/UNIX 1y 54.0 0.0070 1 2011-11-08T18:03:20+0000 default active USD Medium
Utilization
RESERVEDINSTANCES d16f7a91-556f-4db5-afc9-4dd0673334c6 us-east-1a m1.large
Windows with SQL Server 3y 3400.0 0.35 1 2011-12-02T06:22:03+0000 default
active USD Medium Utilization
REQUEST ID d9121e8b-e7c1-49f2-88cd-b478cce99751
```

- API

Use the `DescribeReservedInstances` action to find out the pricing tier in which your account belongs. The `DescribeReservedInstances` action will return `fixedPrice` information showing the amount you paid for all your active Reserved Instances.

Your `DescribeReservedInstances` action should look like the following example:

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

Amazon EC2 returns output similar to the following example:

```
<DescribeReservedInstances xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">

 <reservedInstancesSet>
 <item>
 <reservedInstancesId>f127bd27-f0e9-43bb-89f5-1b8c030bc08f9</reserved
InstancesId>
 <instanceType>m1.small</instanceType>
 <availabilityZone>us-east-1b</availabilityZone>
 <start>2011-11-28T16:17:12.845Z</start>
 <duration>31536000</duration>
 <fixedPrice>227.5</fixedPrice>
 <usagePrice>0.03</usagePrice>
 <instanceCount>1</instanceCount>
 <productDescription>Linux/UNIX</productDescription>
 <state>active</state>
 <instanceTenancy>default</instanceTenancy>
 <currencyCode>USD</currencyCode>
 <offeringType>Medium Utilization</offeringType>
 <recurringCharges/>
 </item>
 </reservedInstancesSet>
</DescribeReservedInstances>
```

```
...
</reservedInstancesSet>
</DescribeReservedInstances>
```

## Purchasing at a Discount Tier Price

When you purchase Reserved Instances, Amazon EC2 will automatically apply any discounts to that part of your Reserved Instance purchase that falls within a discounted tier. You don't need to do anything differently, and you can purchase using any of the Amazon EC2 tools.

- AWS Management Console: Click the **Purchase Reserved Instances** button on the **Reserved Instances** page of the [Amazon EC2 console](#).
- Amazon EC2 CLI: Use the `ec2-purchase-reserved-instances-offering` command.
- Amazon EC2 API: Call the `PurchaseReservedInstancesOffering` action.

If your purchase crosses into a discounted pricing tier, the AWS Management Console, the `ec2-describe-reserved-instances` command, or the `DescribeReservedInstances` action will show multiple entries for that purchase. You will see an entry for that part of the purchase that will be charged the regular Reserved Instance price, and another entry or entries for that part of the purchase that will be charged the applicable discounted rate.

For example, let's say that you have a list price total of \$245,000 in Reserved Instances in a region. You purchase more Reserved Instances at a total list price of \$10,000. Remember that the threshold for the 10 percent discount tier is \$250,000, so \$5,000 of your purchase crosses into the discount tier. When the transaction is complete, the list you'll get from the AWS Management Console, the CLI, or the API will show two reservations—a reservation for \$5,000 at the undiscounted rate, and another for \$4,500, which is the discounted price. The discount is 10 percent of \$5,000 or \$500.

The pricing discount applies to future purchases after the total list price cost of your active Reserved Instances reaches the discounted pricing tier. However, if some of your Reserved Instances expire and the total list price for your active Reserved Instances falls below the discounted pricing tier level, then the next purchase you make will be at the retail rates for Reserved Instances. To illustrate, let's use the previous example: You currently have a list price total of \$254,500 in Reserved Instances. \$250,000 is at the undiscounted rate, and \$4,500 is at the discounted rate. In two months, \$20,000 worth of your Reserved Instances expire, and your total is down to \$234,500, which is below the threshold for the 10 percent discount tier. When you purchase \$15,000 in Reserved Instances, you will be charged the retail rate. However, the \$4,500 in Reserved Instances that you purchased earlier at the discounted rate will continue to be charged at the discounted rate.

The main point to keep in mind is that *list price* is the undiscounted price of the Reserved Instance at the time of purchase, while the *fixed price* value is the *paid price* of all Reserved Instances purchased. The discount tier is based on *list price*.

In summary, when you purchase Reserved Instances, one of three possible scenarios will occur:

- Your purchase of Reserved Instances per region is still within the lowest tier (tier 0). This means that your purchase does not reach the threshold for the discount tiers, and you don't get a discount.
- Your purchase of Reserved Instances within a region crosses the threshold of the first discount tier. This means that the newly purchased Reserved Instances will go into the pending state while the purchase is processed. For this purchase, the Reserved Instances service will purchase Reserved Instances for you at two different rates: An amount at the undiscounted rate, and an amount at the discounted rate. It is important to understand that the Reserved Instance IDs you get back from the CLI or API will be different from the new Reserved Instance IDs that will actually be created. For an explanation of this difference, see the next section, [Reserved Instance IDs \(p. 234\)](#).

- Your entire purchase of Reserved Instances within a region is completely within a discount tier. This means that the newly purchased Reserved Instances will go into the pending state while the purchase is processed. For this purchase, the Reserved Instances service will purchase Reserved Instances at the appropriate discount level for you. It is important to understand that, as with the previous scenario, the Reserved Instance IDs you get back from the CLI or API will be different from the new Reserved Instance IDs that will actually be created. For an explanation of this difference, see the next section, [Reserved Instance IDs \(p. 234\)](#).

## Reserved Instance IDs

When your total purchase crosses one or more Reserved Instance discount pricing tiers, the Reserved Instance IDs returned by your purchase command can be different from the Reserved Instance IDs returned by the describe command that you call after the purchase is complete. What happens is that the Reserved Instance service generates several Reserved Instance IDs because your purchase crossed from an undiscounted tier to a discounted tier, or from one discounted tier to another. There will be a Reserved Instance ID for each set of Reserved Instances in a tier.

Using the CLI, here's an example output of the `ec2-describe-reserved-instances` command if you use the Reserved Instance ID `1ba8e2e3-edf1-43c3-b587-7742bc77b9ba`, which was returned by `ec2-purchase-reserved-instances-offering`. The describe command will not recognize the Reserved Instance ID (`1ba8e2e3-edf1-43c3-b587-7742bc77b9ba`).

```
$ ec2-describe-reserved-instances -H --region sa-east-1 1ba8e2e3-edf1-43c3-b587-7742bc77b9ba
Type ReservedInstancesId AvailabilityZone InstanceType ProductDescription Duration FixedPrice UsagePrice InstanceCount Start State Currency InstanceTenancy OfferingType
```

In this example, the Reserved Instance service generated several Reserved Instance IDs because your purchase crossed from the undiscounted tier (tier 0) to the first discounted tier (tier 1). The portion of your Reserved Instances purchase that is in tier 1 has a 10 percent discount applied to the list price.

Your entire purchase would look like the following example:

```
$ ec2-describe-reserved-instances -H --region sa-east-1 bbcd9749-05f0-4ada-96c8-812f5f0ab9b3
Type ReservedInstancesId AvailabilityZone InstanceType ProductDescription Duration FixedPrice UsagePrice InstanceCount Start State Currency InstanceTenancy OfferingType
RESERVEDINSTANCES bbcd9749-05f0-4ada-96c8-812f5f0ab9b3 sa-east-1a t1.micro
Linux/UNIX 3y 20000.0 0.0090 2 2012-03-02T23:20:16+0000 default payment-pending
USD Medium Utilization
$ ec2-describe-reserved-instances -H --region sa-east-1 1ba8e2e3-346e-4e5b-a2e2-b559243f2325
Type ReservedInstancesId AvailabilityZone InstanceType ProductDescription Duration FixedPrice UsagePrice InstanceCount Start State Currency InstanceTenancy OfferingType
RESERVEDINSTANCES 1ba8e2e3-346e-4e5b-a2e2-b559243f2325 sa-east-1a t1.micro
Linux/UNIX 3y 18000.0 0.0080 3 2012-03-02T23:20:17+0000 default payment-pending
USD Medium Utilization
$ ec2-describe-reserved-instances -H --region sa-east-1 af9f760e-868c-48f4-87e2-44576dbf05ef
Type ReservedInstancesId AvailabilityZone InstanceType ProductDescription Duration FixedPrice UsagePrice InstanceCount Start State Currency InstanceTenancy OfferingType
RESERVEDINSTANCES af9f760e-868c-48f4-87e2-44576dbf05ef sa-east-1a t1.micro
```

```
Linux/UNIX 3y 18000.0 0.0080 5 2012-03-02T23:20:18+0000 default payment-pending
USD Medium Utilization
```

Reserved Instance ID `bbcd9749-05f0-4ada-96c8-812f5f0ab9b3` is the ID for the Reserved Instances that you purchased at the undiscounted list price of \$20,000 each. Reserved Instance IDs `1ba8e2e3-346e-4e5b-a2e2-b559243f2325` and `af9f760e-868c-48f4-87e2-44576dbf05ef` are the IDs for the Reserved Instances that you purchased at the 10 percent discount rate (\$20,000 minus \$2,000).

### Scenario Showing Purchases that Cross Pricing Tiers

Let's walk through an example scenario in which your purchases cross the various pricing tiers.

Two months ago, you purchased 100 Reserved Instances in the us-east-1a Region at \$2000 each. That purchase totaled \$200,000. The list price for this purchase is \$2000. The amount you paid was \$2000 per Reserved Instance, so it is the paid price, and the paid price is the value that will be shown under fixed price. Your purchases are still within the first, undiscounted tier. (For information about tier thresholds, see the [Reserved Instance Pricing Tiers \(p. 230\)](#) table.)

The following table illustrates this example.

Purchase Number	List Price	Amount Paid	Fixed Price	Total RI Purchased	Total List Price Cost	Total Paid Amount
Purchase 1	\$2000	\$2000	\$2000	100	\$200,000	\$200,000

Later you want to purchase more Reserved Instances. Let's say that AWS lowered prices and the same type of Reserved Instance now is available at \$1000 each. You purchase 75 of these Reserved Instances. The list price for this purchase is \$1000. The purchase of 75 Reserved Instances at \$1000 each totals \$75,000. This raises the total cost of your active Reserved Instances to \$275,000. The threshold for the discount tier is \$250,000. This purchase crosses you into the first discount tier, tier 1 in the [Reserved Instance Pricing Tiers \(p. 230\)](#) table.

In this discount tier, you get a 10 percent discount on all your purchases in the same region of Reserved Instances above \$250,000. So, you will pay the new list price of \$1000 each for the first 50 Reserved Instances (total amount paid of \$50,000). And you will pay \$900 each—the \$1000 list price, minus the 10 percent discount—for the remaining 25 Reserved Instances (total amount paid of \$22,500). Your *fixed price* and the *amount paid* for the discounted Reserved Instances will both show \$900.

The following table illustrates this example.

Purchase Number	List Price	Amount Paid	Fixed Price	Total RI Purchased	Total List Price Cost	Total Paid Amount
Purchase 1	\$2000	\$2000	\$2000	100	\$200,000	\$200,000
Purchase 2	\$1000	\$1000	\$1000	50	\$50,000	\$50,000
	\$1000	\$900	\$900	25	\$25,000	\$22,500
Totals					\$275,000	\$272,500

Six months later, let's assume that your business has experienced tremendous growth, and you need to purchase 1800 additional Reserved Instances in the same region. At a list price of \$1000, this purchase will total \$1,800,000. When you add this new purchase to your previous purchases of already active Reserved Instances in the amount of \$272,500, your new total will be \$2,072,500. This new total crosses

the threshold for the next discount tier (tier 2 in the [Reserved Instance Pricing Tiers \(p. 230\)](#) table). In this tier, discounts of 20 percent apply to purchases above \$2,000,000.

Your new purchase will be charged two different discount rates: The 1725 Reserved Instances that fall within tier 1 will be discounted at 10 percent. The remaining 75 Reserved Instances that put the total list price cost above \$2,000,000, and thus are in tier 2, will be discounted at 20 percent.

Purchase Number	List Price	Amount Paid	Fixed Price	Total RI Purchased	Total List Price Cost	Total Paid Amount
Purchase 1	\$2000	\$2000	\$2000	100	\$200,000	\$200,000
Purchase 2	\$1000	\$1000	\$1000	50	\$50,000	\$50,000
	\$1000	\$900	\$900	25	\$25,000	\$22,500
Purchase 3	\$1000	\$900	\$900	1725	\$1,725,000	\$1,552,500
	\$1000	\$800	\$800	75	\$75,000	\$60,000
				Total	\$2,075,000	\$1,885,000

## Reading Your Bill

A bill for a Reserved Instances purchase that qualifies for a discount shows the split purchase if your purchase crosses a pricing tier. Your bill will reflect a breakdown of costs similar to the tiered price scenario discussed in the previous section. In the previous example, you save \$390,000 (the difference between the total list price cost and the total paid amount) using Reserved Instances pricing tiers.

## Current Limitations

The following limitations currently apply:

- Reserved Instance ID: If your Reserved Instance purchase crosses into a discount pricing tier or crosses between discount pricing tiers, your new purchase will go into a pending state while the Reserved Instance service processes the purchase. Processing involves purchasing two or more different Reserved Instances for you: An amount at either the undiscounted rate or the lower discounted rate, and an amount at either the lower discounted rate or the higher discounted rate.
- Describe Reserved Instances: The Reserved Instance ID returned by your purchase CLI command and API action will be different from the actual ID of the new Reserved Instances.
- Amazon EC2 Reserved Instance purchases are the only purchases that will apply toward your Amazon EC2 Reserved Instance pricing tiers. And the Amazon EC2 Reserved Instance pricing tiers and related discounts apply only to purchases of Amazon EC2 Reserved Instances.
- Amazon EC2 Reserved Instance pricing tiers do not apply to instances running Microsoft SQL Server.
- Amazon EC2 Reserved Instances purchased as part of a tiered discount cannot be sold in the Reserved Instance Marketplace. For more information about the Reserved Instance Marketplace, see [Reserved Instance Marketplace \(p. 243\)](#).

For a checklist that summarizes requirements for working with Reserved Instances and the Reserved Instance Marketplace, see [Requirements Checklist for Reserved Instances \(p. 272\)](#).

# Obtaining Information About Your Reserved Instances

Information about your Reserved Instances, such as state, instance type, Availability Zone, and term is useful when you decide to use the capacity reservation. You can check information about the Reserved Instances that are available to your account using any of the Amazon EC2 tools that you've used either for purchasing or selling.

## Reserved Instance States

Reserved Instances can be in one of the following states:

- **Active**—The Reserved Instance is available for use.
- **Payment-Pending**—Amazon Web Services (AWS) is processing your payment for the Reserved Instance. You will be able to use the Reserved Instance when the state becomes Active.
- **Retired**—The Reserved Instance has been terminated. It could have reached this state because of any of the following reasons:
  - AWS did not receive your payment. For example, the credit card transaction did not go through.
  - The Reserved Instance term expired.
  - The Reserved Instance was canceled.

Status information displayed in the **State** column in the **Reserved Instance** page is different from the status information displayed in the **Listing State** in the **My Listings** tab. The State column displays status of the *Reserved Instance*; the Listing State displays status of the Reserved Instance *Listing*. The **My Listings** tab shows information only if you are a seller in the Reserved Instance Marketplace. For more information, see [Reserved Instance Listing States \(p. 270\)](#).

## AWS Management Console

### To view your listing

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Reserved Instances** in the **Navigation** pane.

The **Reserved Instances** page displays a list of your account's instances.

3. Select a Reserved Instance. The **Details** tab displays details about the instance you selected.

The screenshot shows the AWS Management Console interface for Reserved Instances. At the top, there are three buttons: "Purchase Reserved Instances", "Purchase More Like These", and "Sell Reserved Instances". Below that is a dropdown menu set to "All Reserved Instances" and a search bar. The main area displays a table of reserved instances with columns: RI ID, Instance Type, Zone, Term, Offering Type, Fixed Price, Usage Price, Instance Count, Recurring Charges, and State. One row is highlighted with a yellow background, indicating it is selected. A red arrow points from the text "1 Reserved Instance selected" to this highlighted row. Below the table, a message says "1 Reserved Instance selected" followed by "Reserved Instance: 46a408c7-9c23-494d-b68e-38c767ca2e3e". There are two tabs at the bottom: "Details" (which is active) and "My Listings". The "Details" tab shows the following information for the selected instance:  
Instance Type: t1.micro  
Instance Count: 1  
Start: 2012-09-11 12:18 PDT  
Expires: 2013-05-09 12:18 PDT  
Fixed Price: \$0.88  
Usage Price: \$0.006  
Availability Zone: us-east-1d  
Description: Linux/UNIX  
Term: 8 months  
Time left: around 1 day  
State: active  
Tenancy: default

4. If you are a seller in the Reserved Instance Marketplace and you want information about your Reserved Instance listing, click the **My Listings** tab. You will see details about the Reserved Instance listing you selected.

RI ID	Instance Type	Zone	Term	Offering Type	Fixed Price	Usage Price	Instance Count	Recurring Charges	State
<input checked="" type="checkbox"/> 46a08c7-08cf-4aec-b6	11 micro	us-east-1b	12 months	Light Utilization	\$23.0	\$0.012	1	-	retired
<input type="checkbox"/> 46a08c7-08cf-4aec-b6	11 micro	us-east-1d	8 months	Medium Utilization	\$0.88	\$0.006	1	-	active
<input type="checkbox"/> 9d5cb137-7251-4a22-8	11 micro	us-east-1d	8 months	Medium Utilization	\$1.96	\$0.006	1	-	active

1 Reserved Instance selected

**Reserved Instance:** 46a08c7-08cf-4aec-b68e-19af54725f29

**Details** | My Listings

Listings: All

**Listing State: Closed**

Inventory Details	Listing Details:
Available: 0	Listing Term:
Pending: 0	Upfront Listing Price:
Sold: 1	Competitive Pricing Benchmark:
Cancelled: 0	Total: 1 Suggested Price: Lowest Comparable Market Listing:

## Amazon EC2 CLI

### To view your listing

- Run `ec2-describe-reserved-instances-listing` to get details about your listing.

```
PROMPT> ec2-describe-reserved-instances-listings
```

Amazon EC2 returns output similar to the following:

```
PROMPT> ec2-describe-reserved-instances-listings
Type ReservedInstancesListingId ReservedInstancesId CreateDate UpdateDate
Status StatusMessage
LISTING 615d8a10-8224-4c19-ba7d-b9aa0example 1ba8e2e3-d20d-44ec-b202-
fcb6aexample Wed Aug 22 09:02:58 PDT 2012 Wed Aug 22 14:24:26 PDT 2012 can
celled cancelled
INSTANCE-COUNT available 0
INSTANCE-COUNT sold 0
INSTANCE-COUNT cancelled 1
INSTANCE-COUNT pending 0
PRICE-SCHEDULE 10 $1.2
PRICE-SCHEDULE 9 $1.08
PRICE-SCHEDULE 8 $0.96
PRICE-SCHEDULE 7 $0.84
PRICE-SCHEDULE 6 $0.72
PRICE-SCHEDULE 5 $0.6
PRICE-SCHEDULE 4 $0.48
PRICE-SCHEDULE 3 $0.36
PRICE-SCHEDULE 2 $0.24
PRICE-SCHEDULE 1 $0.12
LISTING d5fa5166-83c3-40e4-abb2-b7298example 1ba8e2e3-d20d-44ec-b202-
fcb6aexample Wed Aug 22 14:31:55 PDT 2012 Wed Aug 22 14:42:40 PDT 2012 closed
closed
INSTANCE-COUNT available 0
INSTANCE-COUNT sold 1
INSTANCE-COUNT cancelled 0
INSTANCE-COUNT pending 0
PRICE-SCHEDULE 10 $0.9
PRICE-SCHEDULE 9 $0.81
PRICE-SCHEDULE 8 $0.72
```

```
PRICE-SCHEDULE 7 $0.63
PRICE-SCHEDULE 6 $0.54
PRICE-SCHEDULE 5 $0.45
PRICE-SCHEDULE 4 $0.36
PRICE-SCHEDULE 3 $0.27
PRICE-SCHEDULE 2 $0.18
PRICE-SCHEDULE 1 $0.09
...
LISTING 095c0e18-c9e6-4692-97e5-653e0example b847fa93-c736-4eae-bca1-e3147example Tue Aug 28 18:21:07 PDT 2012 Tue Aug 28 18:21:07 PDT 2012 active
active
INSTANCE-COUNT available 1
INSTANCE-COUNT sold 0
INSTANCE-COUNT cancelled 0
INSTANCE-COUNT pending 0
PRICE-SCHEDULE 5 $1.2
PRICE-SCHEDULE 4 $1.2
PRICE-SCHEDULE 3 $1.2
PRICE-SCHEDULE 2 $1.2
PRICE-SCHEDULE 1 $1.2
```

## Amazon EC2 API

### To view your listing

- Call `DescribeReservedInstancesListings` to get details about your listing.

The call should look like this example:

```
http://ec2.amazonaws.com/?Action=DescribeReservedInstancesListings
&AUTHPARAMS
```

Following is an example response.

```
<DescribeReservedInstancesListingsResponse>
 <requestId>cec5c904-8f3a-4de5-8f5a-ff7f9example</requestId>
 <reservedInstancesListingsSet>
 <item>
 <reservedInstancesListingId>5ec28771-05ff-4b9b-aa31-9e57dexample</reservedInstancesListingId>
 <reservedInstancesId>f127bd27-cee4-443a-a76b-a5af9example</reservedInstancesId>
 <createDate>2012-08-30T17:11:09.449Z</createDate>
 <updateDate>2012-08-30T21:00:42.300Z</updateDate>
 <status>active</status>
 <statusMessage>active</statusMessage>
 <instanceCounts>
 <item>
 <state>Available</state>
 <instanceCount>2</instanceCount>
 </item>
 <item>
 <state>Sold</state>
 <instanceCount>1</instanceCount>
```

```
</item>
<item>
 <state>Cancelled</state>
 <instanceCount>0</instanceCount>
</item>
<item>
 <state>Pending</state>
 <instanceCount>0</instanceCount>
</item>
</instanceCounts>
<priceSchedules>
 <item>
 <term>11</term>
 <price>2.5</price>
 <currencyCode>USD</currencyCode>
 <active>true</active>
 </item>
 <item>
 <term>10</term>
 <price>2.5</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>9</term>
 <price>2.5</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>8</term>
 <price>2.0</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>7</term>
 <price>2.0</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>6</term>
 <price>2.0</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>5</term>
 <price>1.5</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>4</term>
 <price>1.5</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
```

```
</item>
<item>
 <term>3</term>
 <price>0.7</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
</item>
<item>
 <term>2</term>
 <price>0.7</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
</item>
<item>
 <term>1</term>
 <price>0.1</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
</item>
</priceSchedules>
<tagSet/>
<clientToken>listRI1</clientToken>
</item>
</reservedInstancesListingsSet>
</DescribeReservedInstancesListingsResponse>
```

## Understanding the Pricing Benefit of Reserved Instances

When you purchase Reserved Instances, you get two benefits: a capacity reservation for a number of EC2 instances you will launch at some future time, and a discounted hourly fee. The rest of the experience of launching and working with Reserved Instances is the same as working with On-Demand EC2 instances.

This section discusses how Amazon EC2 applies the pricing benefit of Reserved Instances.

### Applying the Pricing Benefit of Reserved Instances

With Reserved Instances, you pay an upfront fee for the capacity reservation on available Amazon EC2 instances based on the specifications (such as product platform, instance type, Availability Zone, etc.) that you defined during the purchase. After you purchase Reserved Instances, you cannot change these specifications. For example, if you purchased a c1.medium instance, you cannot change the capacity reservation to a c1.xlarge instance. (However, you can sell your unused Reserved Instances in the Reserved Instance Marketplace.)

In addition to the capacity reservation, you also get a discounted rate on the hourly fee for running On-Demand EC2 instances that are associated with the same account that purchased the Reserved Instances. For the discount to apply, the On-Demand instances must match the specifications for the Reserved Instances.

For example, let's say user A is running the following ten On-Demand EC2 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

Then user A purchases the following six Medium Utilization Reserved Instances:

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

When he purchases the Reserved Instances, user A pays an upfront fee for the capacity reservation so he can launch the six instances to his specifications when he needs them. In addition, he gets a discount on the hourly usage fees for the equivalent of six instances each month. Since he already has instances running when he purchases the Reserved Instances, Amazon EC2 will automatically apply the discounted hourly rates to the already running On-Demand instances that match the specifications for the Reserved Instances he purchased.

This is what happens:

- Amazon EC2 applies the discounted usage fee rate for two m1.small Reserved Instances that user A purchased to two of the four running m1.small Amazon EC2 instances in Availability Zone us-east-1a.

The other two EC2 instances in Availability Zone us-east-1a will be charged at the current On-Demand rate.

- Amazon EC2 doesn't apply discounted rates from the three c1.medium Reserved Instances that user A purchased because these c1.medium Reserved Instances are specified to run in a different Availability Zone from the zone currently running c1.medium Amazon EC2 instances.

The four running c1.medium Amazon EC2 instances will be charged at the current On-Demand rate.

If user A launches a c1.medium EC2 instance in Availability Zone us-east-1a, then Amazon EC2 will apply the Reserved Instance discounted usage fee rate to that instance.

- Amazon EC2 applies the discounted usage fee rate for one c1.xlarge Reserved Instance that user A purchased to one of the two running c1.xlarge Amazon EC2 instances in Availability Zone us-east-1b.

The other c1.xlarge EC2 instance in Availability Zone us-east-1b will be charged at the current On-Demand rate.

In this example scenario, by purchasing the six Reserved Instances, user A saves on the hourly fee charged against two m1.small and one c1.xlarge On-Demand EC2 instances he had already running. At the same time, he is assured of the capacity to run the six Reserved Instances when he needs them.

## Reserved Instances and Consolidated Billing

The pricing benefits of Reserved Instances are shared when the purchasing account is part of a set of accounts billed under one consolidated billing (CB) payer account. Consolidated billing allows you to pay all of your charges using one account, the CB payer account. The amount of hourly usage for each month across all sub-accounts is also aggregated in the CB payer account. This billing is typically useful for companies in which there are different functional teams or groups. For more information on consolidated billing, see *Consolidated Billing* in [About AWS Account Billing](#).

For Reserved Instances, the amount of usage across all linked accounts is aggregated in the CB payer account, by the hour for each month. Then the normal Reserved Instance logic is applied to calculate the bill.

For example, your account is part of a consolidated billing account, and using your account you purchase Reserved Instances. The upfront cost of the Reserved Instances is paid by the CB payer account, and the discount is spread across the sub-accounts. The allocation of the total cost is determined by the ratio of each sub-account's usage divided by the total usage of the CB payer account. However, the capacity reservation remains with the sub-account that purchased the reservation—in this example, your account.

Keep in mind that capacity reservation only applies to the product platform, instance type, and Availability Zone specified in the purchase.

For more information about how the discounts of the Reserved Instance pricing tiers apply to consolidated billing accounts, see [Understanding Discount Pricing Tiers \(p. 229\)](#).

## Reserved Instance Marketplace

The Reserved Instance Marketplace is an online marketplace that provides AWS customers the flexibility to sell their unused Amazon Elastic Compute Cloud (Amazon EC2) Reserved Instances to other businesses and organizations. Customers can also browse the Reserved Instance Marketplace to find a wide selection of Reserved Instance term lengths and pricing options sold by other AWS customers (listed as *3rd-Party sellers*). The Reserved Instance Marketplace gives customers the flexibility to sell the remainder of their Reserved Instances as their needs change. For example, a customer may want to move instances to a new AWS region, change to a new instance type, or sell capacity for projects that end before the term expires. Amazon EC2 Instances purchased on the Reserved Instance Marketplace offer the same capacity reservations as Reserved Instances purchased directly from AWS.

### Note

Some restrictions—such as what is required to become a seller and when you can sell your reserved capacity—apply. For information about restrictions and requirements for Reserved Instances and the Reserved Instance Marketplace, see [Requirements Checklist for Reserved Instances \(p. 272\)](#).

For a buyer, there are a few differences between these Reserved Instances and Reserved Instances purchased directly from Amazon Web Services (AWS):

- **Term.** The Reserved Instances that you purchase from third-party sellers on the Reserved Instance Marketplace will have less than a full standard term remaining. Full standard terms for Reserved Instances available from AWS run for one year or three years.
- **Upfront price.** Third-party Reserved Instances can be sold at different upfront prices. The usage or recurring fees will remain the same as the fees set when the Reserved Instances were originally purchased from AWS.
- **Tiered discounts.** Amazon EC2 Reserved Instances purchased at a reduced cost because a discount tier threshold had been crossed cannot be sold in the Reserved Instance Marketplace. For information about the Reserved Instance pricing tiers, see [Understanding Discount Pricing Tiers \(p. 229\)](#)

As a seller, you can choose to list some or all of your Reserved Instances, and you can choose the upfront price you want to receive. Your Reserved Instances are then listed in the Reserved Instance Marketplace and are available for purchase. You will be charged a service fee of 12 percent of the total upfront price for each Reserved Instance you sell in the Reserved Instance Marketplace. You can use your Reserved Instance until it's sold. When you sell, you are giving up the capacity reservation and the accompanying discounted fees. This means that you can continue to use your instance after you have sold your capacity reservation. You will just have to pay the On-Demand price for the instance, starting from the time that the reserved capacity on the instance was sold.

### Note

Only Amazon EC2 Reserved Instances can be sold in the Reserved Instance Marketplace. Other AWS Reserved Instances, such as Amazon Relational Database Service (Amazon RDS) and Amazon ElastiCache Reserved Instances cannot be sold on the Reserved Instance Marketplace.

For information about Reserved Instances, see [Reserved Instances \(p. 212\)](#).

## Buyer Overview

The Reserved Instance Marketplace is useful if you want to buy Reserved Instances with terms that are different from the terms offered by AWS. You can search the marketplace for Reserved Instances with configurations that address your specific business needs.

### Quick Start: Buying in the Reserved Instance Marketplace Video

The following video shows you how to buy Reserved Instances in the Reserved Instance Marketplace using the AWS Management Console. [Getting Started Buying Reserved Instances in the Reserved Instance Marketplace](#)

#### Requirements

To purchase Reserved Instances in the Reserved Instance Marketplace, you must have a valid Amazon Web Services (AWS) account. For information on setting up an AWS account, see [Getting Started with Amazon EC2 Linux Instances \(p. 8\)](#).

If you have purchased Amazon EC2 Reserved Instances in the past, you will find that the process and tools for purchasing Reserved Instances in the Reserved Instance Marketplace are very familiar.

#### Steps to buying Reserved Instances

The steps to purchasing Reserved Instances—whether they are standard AWS Reserved Instances or instances in the Reserved Instance Marketplace—are the same.

1. Specify the details of the Reserved Instance you want to purchase.
2. Select the Reserved Instance you want from the list identified by the Reserved Instance Marketplace based on your specifications.
3. Confirm your choice and purchase.

For more information, see [Purchasing Reserved Instances \(p. 221\)](#).

## Seller Overview

The Reserved Instance Marketplace will be useful to you if you own Reserved Instances and want to sell the remainder of the term of your reserved capacity, or if your business is looking for Reserved Instances with configurations that are different from the ones you currently own. The marketplace gives you the opportunity to sell your instances to businesses with needs for short-term workloads and they want to purchase Reserved Instances outside the standard one-year and three-year term lengths.

Listing on the Reserved Instance Marketplace provides you the flexibility to move to new Reserved Instance configurations when your business needs change. For example, say you currently own several three-year, m1.xlarge Reserved Instances in the EU (Ireland) Region. This year, your customer base expanded to Asia, so you need an m1.large Reserved Instance that you can use in the Asia Pacific (Tokyo) Region. You can use the Reserved Instance Marketplace to sell the remainder of the term on some of your m1.xlarge Reserved Instances purchased in the EU (Ireland) Region and purchase capacity in the Asia Pacific (Tokyo) Region.

#### Note

AWS will charge you a service fee of 12 percent of the total upfront price of each Reserved Instance you sell in the marketplace.

### Quick Start: Selling in the Reserved Instance Marketplace Video

The following video shows you how to sell Reserved Instances in the Reserved Instance Marketplace using the AWS Management Console. This video includes instructions on registering as a seller and listing your instances. [Getting Started Selling Reserved Instances in the Reserved Instance Marketplace](#)

## Requirements

- **Register as a seller.** Any US legal entity or non-US legal entity can sell in the Reserved Instance Marketplace by first registering as a seller. For information, see [Registering as a Seller \(p. 246\)](#).
- **Complete the tax registration.** Sellers who have 200 or more transactions or who plan to sell \$20,000 or more in Reserved Instances will have to provide additional information about their business for tax reasons. For information, see [Tax Information \(p. 248\)](#).
- **Provide a US bank.** AWS must have your bank information in order to disburse funds collected when you sell your Reserved Instance. The bank you specify must have a US address. For more information, see [Your Bank \(p. 247\)](#).

## Steps to selling Reserved Instances

After you have registered as a seller and have provided all required information, you are ready to sell your Reserved Instances in the Reserved Instance Marketplace.

1. Select the Reserved Instances you want to sell.
2. Choose the price at which you want your Reserved Instances to sell.
3. List your Reserved Instances.

For more information, see [Selling in the Reserved Instance Marketplace \(p. 245\)](#).

## What Do You Want to Do Next?

- Learn about:
  - [Getting Started with Reserved Instances \(p. 213\)](#)
  - [Understanding the Pricing Benefit of Reserved Instances \(p. 241\)](#)
  - [Understanding Discount Pricing Tiers \(p. 229\)](#)
- Start:
  - [Obtaining Information About Your Reserved Instances \(p. 237\)](#)
  - [Purchasing Reserved Instances \(p. 221\)](#)
  - [Registering as a Seller \(p. 246\)](#)
  - [Listing Your Reserved Instance \(p. 251\)](#)

## Selling in the Reserved Instance Marketplace

The Reserved Instance Marketplace gives you the flexibility to sell the remainder of your Reserved Instances as your needs change—for example, if you want to move instances to a different Amazon Web Services (AWS) region, change to another instance type, or sell capacity for projects that end before the Reserved Instance term expires. (Some restrictions—such as what is required to become a seller and when you can sell your reserved capacity apply. For information about restrictions and requirements for Reserved Instances and the Reserved Instance Marketplace, see [Requirements Checklist for Reserved Instances \(p. 272\)](#).)

As soon as you list your Reserved Instances, they will be included in a list that other AWS customers can view. AWS groups Reserved Instances based on the type of instance being listed, the duration of the term remaining, and the hourly price. This grouping makes it easier for buyers to find the Reserved Instances they want to purchase. From the list, customers choose to purchase the instance that best matches their criteria and decide on the best tradeoff between quoted upfront price and hourly price.

To fulfill a buyer's request, AWS first sells the Reserved Instance with the lowest upfront price in the specified grouping; then it sells the Reserved Instance with the next lowest price, until the buyer's entire

purchase order is fulfilled. AWS processes the transaction and transfers ownership of the Reserved Instance to the buyer. Sellers will receive a cash disbursement for their Reserved Instances through a wire transfer directly into their bank account.

When you sell in the Reserved Instance Marketplace, the buyer's ZIP code and country information will be provided to you through a disbursement report. With this information, you will be able to calculate any necessary tax you need to remit to the government. Your business name (as the seller) will also be provided on the purchase invoice of the buyer. AWS charges an administrative fee (12 percent of the total upfront price) for selling in the Reserved Instance Marketplace. For more information, see *Reserved Instance Marketplace* in the [Amazon EC2 Reserved Instances](#) product page.

You retain control of your Reserved Instance until it's sold. When you sell, what you are giving up is the capacity reservation and the discounted recurring fees. You can continue to use your instance after you have sold the reserved capacity, but AWS will now charge you the On-Demand price. The On-Demand price will start from the time that your Reserved Instance was sold. Thus, if you don't want to be charged On-Demand prices for instances that you use, purchase more reserved capacity or terminate your instances when your capacity reservation is sold (or expires).

This topic walks you through the steps to selling in the Reserved Instance Marketplace:

- [Registering as a Seller \(p. 246\)](#)—Register as a seller and specify a bank that has a US address. If you plan on 200 or more transactions or if you plan to sell \$20,000 or more worth of Reserved Instances over the course of a year, you also have to provide tax information.
- [Selling Your Reserved Instances \(p. 250\)](#)—List an active Reserved Instance that has more than one month left in its term. You also must own the Reserved Instance for longer than a month.
- [After Your Reserved Instance Is Sold \(p. 270\)](#)—Find out when your Reserved Instance is sold, and how you get paid.
- [Quick Start: Selling in the Reserved Instance Marketplace Video](#)—Pick up the information you need to quickly get started selling in the Reserved Instance Marketplace.

For information about buying Reserved Instances in the Reserved Instance Marketplace, see [Buying Reserved Instances \(p. 219\)](#). For basic information about Reserved Instances, see [Reserved Instances \(p. 212\)](#).

## Important Notes About Selling in the Reserved Instance Marketplace

- To become a seller in the Reserved Instance Marketplace, you must register as a seller, and specify a bank that has a US address. For information, see [Registering as a Seller \(p. 246\)](#).
- Reserved Instances can be sold after they have been active for at least 30 days and when AWS has received the upfront payment.
- You can sell up to \$50,000 in Reserved Instances per year. If you need to sell more Reserved Instances, complete the [Request to Raise Sales Limit on Amazon EC2 Reserved Instances](#) form.
- Amazon EC2 Reserved Instances purchased at a reduced cost resulting from tiered discount cannot be sold in the Reserved Instance Marketplace. For more information about discount pricing tiers, see [Understanding Discount Pricing Tiers \(p. 229\)](#).

For a checklist that summarizes requirements for working with Reserved Instances and the Reserved Instance Marketplace, see [Requirements Checklist for Reserved Instances \(p. 272\)](#).

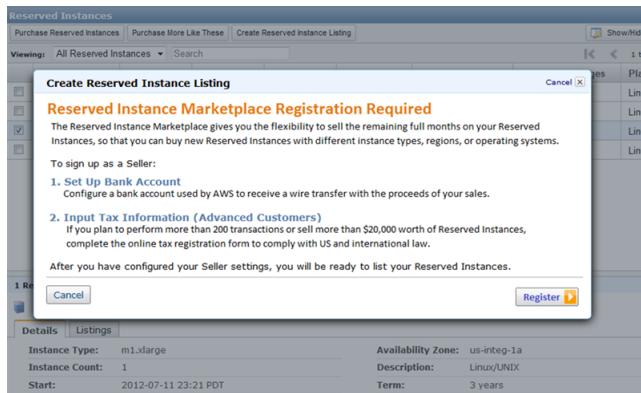
## Registering as a Seller

### Topics

- [Your Bank \(p. 247\)](#)

- [Tax Information \(p. 248\)](#)
- [Seller Registration Confirmation \(p. 249\)](#)
- [Sharing Information with the Buyer \(p. 250\)](#)
- [Next Steps \(p. 250\)](#)

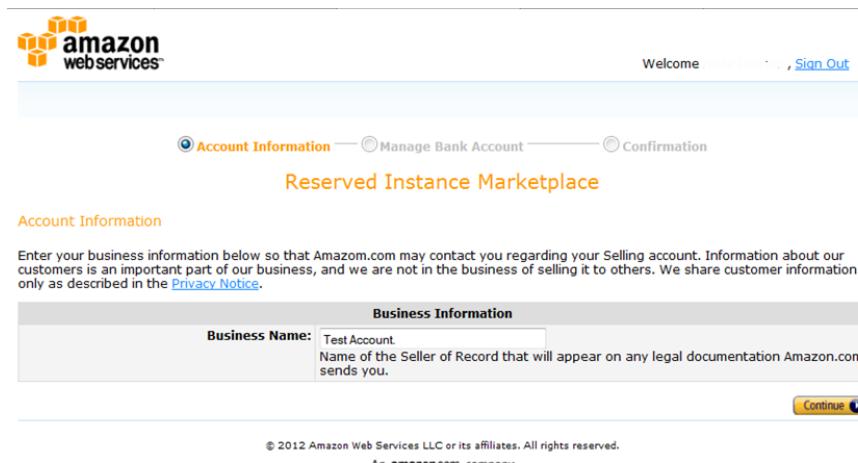
To be able to sell in the Reserved Instance Marketplace, your first task is to register as a seller.



If you haven't created an AWS account yet, you need to do this first before you register for Reserved Instance Marketplace. Complete the instructions described in [Getting Started with Amazon EC2 Linux Instances \(p. 8\)](#), which provides information about creating your Amazon EC2 account and credentials.

You can access the registration process through the [Reserved Instance Marketplace Seller Registration](#) web page. If you try to create a listing and you have not registered, AWS will direct you to this seller registration page.

Registering means providing the name of your business, information about your bank, and your business's tax identification number. Usually, you only have to provide your information once. However, you can update personal and banking information through the [Reserved Instance Marketplace Seller Registration](#) web page. Log in to AWS using the account you used when you first registered and the page will send you directly to the personal and banking information pages.



## Your Bank

AWS must have your bank information in order to disburse funds collected when you sell your Reserved Instance. The bank you specify must have a US address.

On the **Manage Bank Account** page, provide the following information about the bank through which you will receive payment:

- Bank account holder name
- Routing number
- Account number
- Bank account type

Specify if the bank is a personal or corporate account.

- If the bank is your personal account, provide your driver's license number and the state that issued it.
- If you are using a corporate bank account, you will be prompted to send via fax (1-206-765-3424) the information about the bank account.

The screenshot shows the 'Manage Bank Account' page. At the top, there's a navigation bar with tabs: 'Account Information', 'Manage Bank Account' (which is highlighted in orange), and 'Confirmation'. Below the tabs, there's a heading 'Add Bank Account' and a note about registering a U.S. bank account. A sample check stub is displayed, with arrows pointing from the 'Routing Number' and 'Account Number' fields on the form to the corresponding fields on the check stub. The form fields are labeled: 'Bank Account Holder Name:' (with 'Test Account' entered), 'Routing Number:' (with 'us.JK123 (9 digits)' entered), and 'Account Number:' (with 'b.315376745678901234 (3-17 digits)' entered). Below the form, there's a section for selecting a bank account type, with 'Personal Checking Account' selected. There are also fields for 'Driver's License Number' (WA123) and 'State' (WA). At the bottom, there are 'Back' and 'Continue' buttons, along with a link to the 'Privacy Notice'.

You can change the default bank account through which you receive disbursements. Just go to the [Reserved Instance Marketplace Seller Registration](#) web page using the account you used when you first registered and the page will send you directly to the personal and banking information pages.

After you have completed the registration, AWS verifies your bank account and sets it as the default bank. You will not be able to receive disbursements until AWS has verified your account with the bank. Verification with the bank can take up to two weeks, so if your account is a new one, you will not get the disbursement as a result of a sale for up to two weeks. For an established account, it will usually take about two days for disbursements to complete.

## Tax Information

Your sale of Reserved Instances on the Reserved Instance Marketplace might be subject to a transactional tax, such as sales tax or value-added tax. You should check with your business's tax, legal, finance, or

accounting department to determine if transaction-based taxes are applicable. You are responsible for collecting and sending the transaction-based taxes to the appropriate tax authority.

As part of the seller registration process, you have the option of completing a tax interview. We encourage you to complete this process if any of the following apply:

- You want AWS to generate a Form 1099-K.
- You anticipate having either 200 or more transactions or \$20,000 or more in sales of Reserved Instances in a calendar year. A transaction can involve one or more Reserved Instances. If you choose to skip this step during registration, and later you reach transaction 199, you will get a message saying, "You have reached the transaction limit for pre-tax. Please complete the tax interview at [http://portal.aws.amazon.com/ec2/ri/seller\\_registration?action=taxInterview](http://portal.aws.amazon.com/ec2/ri/seller_registration?action=taxInterview) ."
- You are a non-US seller. In this case, you must electronically complete Form W-8BEN.

If you complete the tax interview, the tax information you enter will differ depending on whether your business is a US or non-US legal entity. If you are a US seller, you must provide AWS with your tax identification number along with your business contact information.



After you complete the tax registration process, AWS will file Form 1099-K, and you will receive a copy of it through the US mail on or before January 31 in the year after the year that your tax account reaches the threshold levels.

As you fill out the tax interview, keep in mind the following:

- Information provided by AWS, including the information in this topic, does not constitute tax, legal, or other professional advice. To find out how the IRS reporting requirements might affect your business, or if you have other questions, please contact your tax, legal, or other professional advisor.
- In order to fulfill the IRS reporting requirements as efficiently as possible, answer all questions and enter all information requested during the interview.
- Check your answers. Avoid misspellings or entering incorrect tax identification numbers. They can result in an invalidated tax form.

For more information about IRS requirements and the Form 1099-K, go to the [IRS website](#).

## **Seller Registration Confirmation**

After we receive your completed seller registration, you will get an email confirming your registration and informing you that you can get started selling in the Reserved Instance Marketplace.

## Sharing Information with the Buyer

When you sell your Reserved Instances in the Reserved Instance Marketplace, AWS will share your company's legal name on the buyer's statement to comply with US regulations. In addition, if the buyer calls AWS customer service because the buyer needs to contact you for an invoice or for some other tax-related reason, AWS may need to provide the buyer with your email address so that the buyer can contact you directly.

For similar reasons, the buyer's ZIP code and country information will be provided to the seller in the disbursement report. As a seller you might need this information to accompany any necessary transaction taxes that you remit to the government (such as sales tax and value-added tax).

AWS cannot offer tax advice, but if your tax specialist determines that you need specific additional information, contact AWS Customer Support.

## Next Steps

After you have successfully registered as a seller in the Reserved Instance Marketplace, you can begin selling your Reserved Instances. Selling in the Reserved Instance Marketplace requires the following steps:

1. Deciding which Reserved Instances you want to sell.

Identify the active Reserved Instances that you want to sell and select the upfront price at which you want to sell them. For more information, see [Pricing Your Reserved Instances \(p. 251\)](#).

2. Listing your Reserved Instances.

Include your Reserved Instances in the Reserved Instance Marketplace listings. For more information, see [Listing Your Reserved Instance \(p. 251\)](#).

3. Viewing your listings.

You can monitor your Reserved Instances and view your listings. For more information, see [Obtaining Information About Your Reserved Instances \(p. 237\)](#).

4. Canceling and changing your listings.

You can change your listings by first canceling and then relisting. For more information, see [Canceling and Changing Your Listings \(p. 260\)](#).

For information about selling your Reserved Instances, see [Selling Your Reserved Instances \(p. 250\)](#).

## Selling Your Reserved Instances

### Topics

- [Pricing Your Reserved Instances \(p. 251\)](#)
- [Listing Your Reserved Instance \(p. 251\)](#)
- [Canceling and Changing Your Listings \(p. 260\)](#)

This section walks you through how to list and sell your Reserved Instances in the Reserved Instance Marketplace. If you haven't registered as a seller yet, you must do so first. For information, see [Selling Your Reserved Instances \(p. 250\)](#).

As a registered seller, you can choose to sell one or more of your Reserved Instances, and you can choose to sell all of them in one listing. In addition, you can list any type of Reserved Instance—including any configuration of instance type, platform, region, and Availability Zone—as long as the following requirements are met:

- You have already paid the upfront cost for the Reserved Instance you are listing.

This means that you can list your Reserved Instance when it is in the *Active* state. However, keep in mind that a Reserved Instance can be in the *Active* state before AWS actually receives your payment. If this is the case, the Reserved Instance Marketplace will not allow you to list your Reserved Instance until the payment for the upfront fee is collected.

- You have owned the Reserved Instance for at least a month.
- There is at least a month remaining in the term of the Reserved Instance you are listing.

You can list the remainder of the Reserved Instance term rounded down to the nearest month. For example, if you have 9 months and 13 days remaining on your Reserved Instance, you can list 9 months for your Reserved Instance.

- The Reserved Instance you are selling is not a discounted or private Reserved Instance. You cannot list these types of Reserved Instances in the Reserved Instance Marketplace.

For a checklist that summarizes requirements for working with Reserved Instances and the Reserved Instance Marketplace, see [Requirements Checklist for Reserved Instances \(p. 272\)](#).

To get details about your existing Reserved Instances, you can use any of the following tools:

- The **Reserved Instances** page in the **EC2** console of the [AWS Management Console \(p. 215\)](#).
- The `ec2-describe-reserved-instances` CLI command.
- The `DescribeReservedInstances` API action.

For more information about Reserved Instances, see [Reserved Instances \(p. 212\)](#).

## Pricing Your Reserved Instances

When you're a seller on the Reserved Instance Marketplace, the upfront fee is the only fee that you can specify for the Reserved Instance. The upfront fee is the one-time fee that the buyer pays when the buyer purchases a Reserved Instance. The seller cannot specify the usage fee or the recurring fee. The buyer of your Reserved Instance will be paying the same usage or recurring fees that were set when the Reserved Instances were originally purchased. This usage fee applies to Light Utilization and Medium Utilization Reserved Instances, and the fee differs depending on which Reserved Instance offering you're using. The recurring fee is the hourly fee that you pay for Heavy Utilization, whether or not you're using the Reserved Instance. For more information about the availability of Reserved Instances offerings based on utilization, see [Choosing Reserved Instances Based on Your Usage Plans \(p. 217\)](#).

### Setting a Pricing Schedule

You can set different upfront fees (prices) that are based on when your Reserved Instance sells. You can specify one price if the Reserved Instance sells immediately, and you can specify another price if the Reserved Instance were to sell in any subsequent month. Since the value of Reserved Instances decreases over time, by default, AWS will set prices to decrease linearly—that is, the price drops in equal increments month over month. You can choose to set the prices differently.

For example, if your Reserved Instance has nine months of its term remaining, you can specify the amount you would accept if a customer were to purchase that Reserved Instance with nine months remaining, and you could set another price with five months remaining, and yet another price with one month remaining.

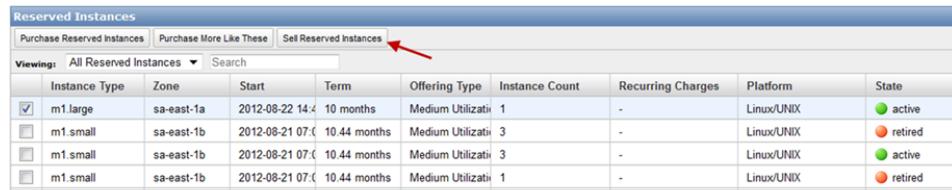
## **[Listing Your Reserved Instance](#)**

You can list the Reserved Instances you want to sell in the Reserved Instance Marketplace by using the AWS Management Console, the Amazon EC2 CLI, or the Amazon EC2 API.

## AWS Management Console

### To list a Reserved Instance in the Reserved Instance Marketplace

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
  2. Click **Reserved Instances** in the **Navigation** pane.
- The **Reserved Instances** page displays a list of your account's instances.
3. Select the Reserved Instances you want to list on the marketplace, and click **Sell Reserved Instances**.



**Reserved Instances**

Instance Type	Zone	Start	Term	Offering Type	Instance Count	Recurring Charges	Platform	State
m1.large	sa-east-1a	2012-08-22 14:4	10 months	Medium Utilizati	1	-	Linux/UNIX	active
m1.small	sa-east-1b	2012-08-21 07:0	10.44 months	Medium Utilizati	3	-	Linux/UNIX	retired
m1.small	sa-east-1b	2012-08-21 07:0	10.44 months	Medium Utilizati	3	-	Linux/UNIX	active
m1.small	sa-east-1b	2012-08-21 07:0	10.44 months	Medium Utilizati	1	-	Linux/UNIX	retired

**1 Reserved Instance selected**

**Reserved Instance:** f127bd27-9f30-41d3-bf45-9af45c7c74da

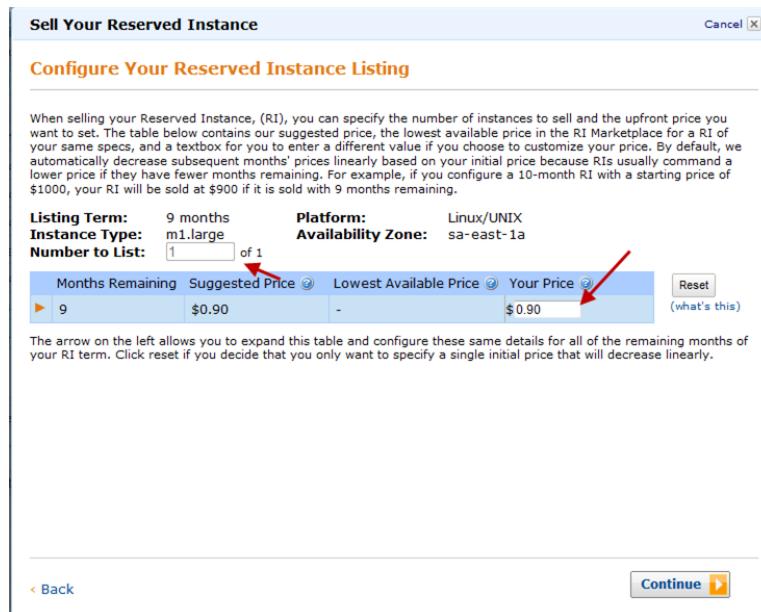
**Details** [ My Listings ]

Instance Type:	m1.large	Availability Zone:	sa-east-1a
Instance Count:	1	Description:	Linux/UNIX
Start:	2012-08-22 14:41 PDT	Term:	10 months
Expires:	2013-06-18 14:41 PDT	Time left:	around 41 weeks 6 days
Fixed Price:	\$1.00	State:	active
Usage Price:	\$0.00	Tenancy:	default

#### Note

If you have not completed the Reserved Instance Marketplace seller registration process, you will be prompted to complete this process now. For information about the seller registration process, see [Registering as a Seller \(p. 246\)](#).

4. In the **Configure Your Reserved Instance Listing** page, for **Number to List**, set the number of instances to sell and, for **Your Price**, set the upfront price for the remaining time period.



**Sell Your Reserved Instance**

**Configure Your Reserved Instance Listing**

When selling your Reserved Instance, (RI), you can specify the number of instances to sell and the upfront price you want to set. The table below contains our suggested price, the lowest available price in the RI Marketplace for a RI of your same specs, and a textbox for you to enter a different value if you choose to customize your price. By default, we automatically decrease subsequent months' prices linearly based on your initial price because RIs usually command a lower price if they have fewer months remaining. For example, if you configure a 10-month RI with a starting price of \$1000, your RI will be sold at \$900 if it is sold with 9 months remaining.

<b>Listing Term:</b> 9 months	<b>Platform:</b> Linux/UNIX		
<b>Instance Type:</b> m1.large	<b>Availability Zone:</b> sa-east-1a		
<b>Number to List:</b> <input type="text" value="1"/> of 1			
Months Remaining	Suggested Price	Lowest Available Price	Your Price
► 9	\$0.90	-	\$0.90

The arrow on the left allows you to expand this table and configure these same details for all of the remaining months of your RI term. Click reset if you decide that you only want to specify a single initial price that will decrease linearly.

[Back](#) [Continue](#)

You can see how the value of your Reserved Instance will change over the remainder of the term by clicking the arrow on the left of the **Months Remaining** column. By default, AWS sets the price to decrease linearly. This means the price drops by equal increments each month.

5. If you are an advanced user and you want to customize the pricing, you can enter different values for the subsequent months. To return to the default linear price drop, click the **Reset** button. Click **Continue** when you are finished configuring your listing.

The screenshot shows the 'Sell Your Reserved Instance' configuration page. In the 'Configure Your Reserved Instance Listing' section, there is a table for setting prices over 9 months for an m1.large instance in sa-east-1a. The table has columns for Months Remaining, Suggested Price, Lowest Available Price, and Your Price. The 'Your Price' column shows a linear decrease from \$0.90 to \$0.30 over 9 months. A note below the table explains that the arrow on the left allows expanding the table to configure details for all remaining months, and a 'Reset' button is available if needed.

Months Remaining	Suggested Price	Lowest Available Price	Your Price
9 months	0.90	-	\$0.90
8 months	0.80	-	\$0.80
7 months	0.70	-	\$0.70
6 months	0.60	-	\$0.60
5 months	0.50	-	\$0.50
4 months	0.40	-	\$0.40
3 months	0.30	-	\$0.30

The arrow on the left allows you to expand this table and configure these same details for all of the remaining months of your RI term. Click reset if you decide that you only want to specify a single initial price that will decrease linearly.

6. When you are satisfied with the details of your listing as displayed by the **Confirm Your Reserved Instance Listing** page, click **List Reserved Instance**.

You will get a confirmation that your listing is being processed.

The screenshot shows the 'Sell Your Reserved Instance' confirmation page. It includes a note about the listing becoming active and a section titled 'Interested in Purchasing New Reserved Instances?'. This section contains two columns: 'Find the Optimal Reserved Instance' and 'Right Sizing Your Instances'. It also includes links to 'Learn more about Reserved Instances' and 'Go to CloudWatch to see your utilization'.

7. To view the details of your Reserved Instance listing, on the **Reserved Instances** page, select the Reserved Instance you want to view, and click the **My Listings** tab.

The screenshot shows the 'Purchase Reserved Instances' section of the Amazon EC2 Reserved Instance Marketplace. At the top, there are three buttons: 'Purchase Reserved Instances', 'Purchase More Like These', and 'Sell Reserved Instances'. Below this is a search bar with the placeholder 'Search' and a dropdown menu set to 'All Reserved Instances'. A table lists three reserved instances:

RI ID	Instance Type	Zone	Term	Offering Type	Fixed Price	Usage Price	Instance Count	Recurring Charges	State
<input checked="" type="checkbox"/> 46a408c7-08cf-4aec-b6-11.micro	11.micro	us-east-1b	12 months	Light Utilization	\$23.0	\$0.012	1	-	retired
<input type="checkbox"/> 46a408c7-08cf-4aec-b6-11.micro	11.micro	us-east-1d	8 months	Medium Utilization	\$0.88	\$0.006	1	-	active
<input type="checkbox"/> 9d5cb137-7251-4a22-9-11.micro	11.micro	us-east-1d	8 months	Medium Utilization	\$1.96	\$0.006	1	-	active

A red arrow points from the 'RI ID' column of the first row to a callout box labeled '1 Reserved Instance selected'. Another red arrow points from the 'State' column of the first row to the same callout box. Below the table, a message says '1 Reserved Instance selected' followed by 'Reserved Instance: 46a408c7-08cf-4aec-b68e-19af54725f29'. There are tabs for 'Details' and 'My Listings', with 'Details' currently selected. A yellow box highlights the 'Listing State: Closed' status. Below this, there are sections for 'Inventory Details' and 'Listing Details', each with several fields like 'Available', 'Pending', 'Sold', 'Cancelled', 'Upfront Listing Price', and 'Suggested Price'. A yellow box highlights the 'Total' field in the 'Inventory Details' section.

## Amazon EC2 CLI

### To list a Reserved Instance in the Reserved Instance Marketplace

- Get a list of your Reserved Instances by calling `ec2-describe-reserved-instances`.

```
PROMPT> ec2-describe-reserved-instances --headers
```

Amazon EC2 returns output similar to the following:

```
PROMPT> ec2-describe-reserved-instances --headers
Type ReservedInstancesId AvailabilityZone InstanceType ProductDescription
Duration FixedPrice UsagePrice InstanceCount Start State Currency Instan
ceTenancy OfferingType
RESERVEDINSTANCES f127bd27-9f30-41d3-bf45-9af45example sa-east-1a m1.large
Linux/UNIX 10m 1.0 0.0 1 2012-08-22T21:41:51+0000 active USD default Medium
Utilization
RESERVEDINSTANCES lba8e2e3-d20d-44ec-b202-fcb6aexample sa-east-1b m1.small
Linux/UNIX 10m 1.2 0.032 3 2012-08-21T14:02:00+0000 retired USD default
Medium Utilization
RESERVEDINSTANCES 4357912c-6f69-4966-a407-6f0cbexample sa-east-1b m1.small
Linux/UNIX 10m 1.2 0.032 3 2012-08-21T14:02:00+0000 active USD default
Medium Utilization
RESERVEDINSTANCES 4357912c-d032-4a97-9b49-5eb3aexample sa-east-1b m1.small
Linux/UNIX 10m 1.2 0.032 1 2012-08-21T14:02:00+0000 retired USD default
Medium Utilization
...
```

Select the Reserved Instance ID of the Reserved Instance you want to list in the Reserved Instance Marketplace.

- Specify the Reserved Instance ID of the Reserved Instance you want to list and call `ec2-create-reserved-instances-listing`. You have to specify the following required parameters:
  - Reserved Instance ID
  - Instance count
  - MONTH:PRICE

The command should look like this example:

```
PROMPT> ec2-create-reserved-instances-listing --reserved-instance b847fa93-c736-4eae-bca1-3147example --instance-count 1 05:01.20 04:01.00 01:00.75 -
-headers
```

Amazon EC2 returns output similar to the following:

```
PROMPT>LISTING 2a0ff720-f62e-4824-8ed1-7dd0aexample b847fa93-c736-4eae-bca1-e3147example Wed Aug 29 13:59:11 PDT 2012 Wed Aug 29 13:59:11 PDT 2012 active
active
INSTANCE-COUNT available 1
INSTANCE-COUNT sold 0
INSTANCE-COUNT cancelled 0
INSTANCE-COUNT pending 0
PRICE-SCHEDULE 5 $1.2
PRICE-SCHEDULE 4 $1.0
PRICE-SCHEDULE 3 $1.0
PRICE-SCHEDULE 2 $1.0
PRICE-SCHEDULE 1 $0.75
```

3. To view the details of your Reserved Instance listing, run `ec2-describe-reserved-instances-listings` with the listing ID `095c0e18-c9e6-4692-97e5-653e0example`.

```
PROMPT> ec2-describe-reserved-instances-listings 095c0e18-c9e6-4692-97e5-
653e0example
```

Amazon EC2 returns output similar to the following:

```
PROMPT> ec2-describe-reserved-instances-listings 095c0e18-c9e6-4692-97e5-
653e0example
Type ReservedInstancesListingId ReservedInstancesId CreateDate UpdateDate
Status StatusMessage
LISTING 095c0e18-c9e6-4692-97e5-653e0example b847fa93-c736-4eae-bca1-
e3147example Tue Aug 28 18:21:07 PDT 2012 Tue Aug 28 18:21:07 PDT 2012 active
active
INSTANCE-COUNT available 1
INSTANCE-COUNT sold 0
INSTANCE-COUNT cancelled 0
INSTANCE-COUNT pending 0
PRICE-SCHEDULE 5 $1.2
PRICE-SCHEDULE 4 $1.2
PRICE-SCHEDULE 3 $1.2
PRICE-SCHEDULE 2 $1.2
PRICE-SCHEDULE 1 $1.2
```

## [Amazon EC2 API](#)

### **To list a Reserved Instance in the Reserved Instance Marketplace**

1. Get a list of your Reserved Instances by calling `DescribeReservedInstances`.

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

Following is an example response.

```
<DescribeReservedInstancesResponse xmlns='http://ec2.amazonaws.com/doc/2012-08-15/'>
<requestId>ebe3410a-8f37-441d-ae11-2e78eexample</requestId>
<reservedInstancesSet>
 <item>
 <reservedInstancesId>f127bd27-cee4-443a-a76b-a5af9example</reservedInstancesId>
 <instanceType>m1.large</instanceType>
 <availabilityZone>us-east-1a</availabilityZone>
 <start>2012-08-07T15:19:31.071Z</start>
 <duration>31536000</duration>
 <fixedPrice>276.0</fixedPrice>
 <usagePrice>0.156</usagePrice>
 <instanceCount>5</instanceCount>
 <productDescription>Linux/UNIX</productDescription>
 <state>active</state>
 <instanceTenancy>default</instanceTenancy>
 <currencyCode>USD</currencyCode>
 <offeringType>Light Utilization</offeringType>
 <recurringCharges/>
 </item>
</reservedInstancesSet>
</DescribeReservedInstancesResponse>
```

Note the Reserved Instance ID of the Reserved Instance that you want to list in the Reserved Instance Marketplace.

2. Create a listing for three Reserved Instances from Reserved Instance ID *f127bd27-cee4-443a-a76b-a5af9example* and specify the following pricing schedule.

Term (remaining months)	11	10	9	8	7	6	5	4	3	2	1
Price specified for period	2.5			2.0			1.5		0.7		0.1
Price	2.5	2.5	2.5	2.0	2.0	2.0	1.5	1.5	0.7	0.7	0.1

The call should look like this example:

```
https://ec2.amazonaws.com/?Action=CreateReservedInstancesListing
&ClientToken=myIdempToken1
&ReservedInstancesId=f127bd27-cee4-443a-a76b-a5af9example
&InstanceCount=3
&PriceSchedules.0.Price=2.5&PriceSchedules.0.Term=11
&PriceSchedules.1.Price=2.0&PriceSchedules.1.Term=8
&PriceSchedules.2.Price=1.5&PriceSchedules.2.Term=5
&PriceSchedules.3.Price=0.7&PriceSchedules.3.Term=3
&PriceSchedules.4.Price=0.1&PriceSchedules.4.Term=1
```

```
&AUTHPARAMS
```

Following is an example response.

```
<CreateReservedInstancesListingResponse>
<requestId>a42481af-335a-4e9e-b291-bd18dexample</requestId>
<reservedInstancesListingsSet>
 <item>
 <reservedInstancesListingId>5ec28771-05ff-4b9b-aa31-
9e57dexample</reservedInstancesListingId>
 <reservedInstancesId>f127bd27-cee4-443a-a76b-a5af9example</reserved
InstancesId>
 <createDate>2012-08-30T17:11:09.449Z</createDate>
 <updateDate>2012-08-30T17:11:09.468Z</updateDate>
 <status>active</status>
 <statusMessage>active</statusMessage>
 <instanceCounts>
 <item>
 <state>Available</state>
 <instanceCount>3</instanceCount>
 </item>
 <item>
 <state>Sold</state>
 <instanceCount>0</instanceCount>
 </item>
 <item>
 <state>Cancelled</state>
 <instanceCount>0</instanceCount>
 </item>
 <item>
 <state>Pending</state>
 <instanceCount>0</instanceCount>
 </item>
 </instanceCounts>
 <priceSchedules>
 <item>
 <term>11</term>
 <price>2.5</price>
 <currencyCode>USD</currencyCode>
 <active>true</active>
 </item>
 <item>
 <term>10</term>
 <price>2.5</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>9</term>
 <price>2.5</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>8</term>
```

```
<price>2.00</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>7</term>
<price>2.0</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>6</term>
<price>2.0</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>5</term>
<price>1.5</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>4</term>
<price>1.5</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>3</term>
<price>0.7</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>2</term>
<price>0.7</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>1</term>
<price>0.1</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
</priceSchedules>
<tagSet/>
<clientToken>listRI1</clientToken>
</item>
</reservedInstancesListingsSet>
</CreateReservedInstancesListingResponse>
```

3. To view the details of your Reserved Instance listing, run `DescribeReservedInstancesListings`.

The command should look like this example:

```
http://ec2.amazonaws.com/?Action=DescribeReservedInstancesListings
&AUTHPARAMS
```

Following is an example response.

```
<DescribeReservedInstancesListingsResponse>
 <requestId>cec5c904-8f3a-4de5-8f5a-ff7f9example</requestId>
 <reservedInstancesListingsSet>
 <item>
 <reservedInstancesListingId>5ec28771-05ff-4b9b-aa31-
9e57dexample</reservedInstancesListingId>
 <reservedInstancesId>f127bd27-cee4-443a-a76b-a5af9example</re-
servedInstancesId>
 <createDate>2012-08-30T17:11:09.449Z</createDate>
 <updateDate>2012-08-30T17:11:09.468Z</updateDate>
 <status>active</status>
 <statusMessage>active</statusMessage>
 <instanceCounts>
 <item>
 <state>Available</state>
 <instanceCount>3</instanceCount>
 </item>
 <item>
 <state>Sold</state>
 <instanceCount>0</instanceCount>
 </item>
 <item>
 <state>Cancelled</state>
 <instanceCount>0</instanceCount>
 </item>
 <item>
 <state>Pending</state>
 <instanceCount>0</instanceCount>
 </item>
 </instanceCounts>
 <priceSchedules>
 <item>
 <term>11</term>
 <price>2.5</price>
 <currencyCode>USD</currencyCode>
 <active>true</active>
 </item>
 <item>
 <term>10</term>
 <price>2.5</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>9</term>
 <price>2.5</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>8</term>
 <price>2.0</price>
```

```
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>7</term>
 <price>2.0</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>6</term>
 <price>2.0</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>5</term>
 <price>1.5</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>4</term>
 <price>1.5</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>3</term>
 <price>0.7</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>2</term>
 <price>0.7</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>1</term>
 <price>0.1</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
</priceSchedules>
<tagSet/>
<clientToken>listRI1</clientToken>
</item>
</reservedInstancesListingsSet>
</DescribeReservedInstancesListingsResponse>
```

## Cancelling and Changing Your Listings

After listing your Reserved Instances in the Reserved Instance Marketplace, you can manage your listing using the [AWS Management Console \(p. 261\)](#), the [Amazon EC2 CLI \(p. 262\)](#), or the [Amazon EC2 API \(p. 264\)](#).

You can view details of your listing. You also can cancel your listing, as long as it hasn't been purchased yet.

Currently, you cannot modify your listing directly. However, you can change your listing by first canceling it and then creating another listing with new parameters.

In this section, we will show you how to perform the tasks of viewing, canceling and changing your Reserved Instance Marketplace listings, using the console, the CLI, and the API.

## AWS Management Console

### To view your listing

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Reserved Instances** in the **Navigation** pane.
3. Right-click your Reserved Instance.
4. Click the **My Listings** tab. You will see details about the Reserved Instance listing you selected.

The screenshot shows the 'Purchase Reserved Instances' page with a table of reserved instances. One instance is selected, indicated by a checked checkbox. Below the table, a message says '1 Reserved Instance selected'. The 'My Listings' tab is selected. In the details panel, the 'Listing State' is shown as 'Closed'. A red arrow points from the 'Selected' checkbox in the table to the 'Listing State' button.

RI ID	Instance Type	Zone	Term	Offering Type	Fixed Price	Usage Price	Instance Count	Recurring Charges	State
45a408c7-08cf-4aec-b6	1t micro	us-east-1b	12 months	Light Utilization	\$23.0	\$0.012	1	-	retired
45a408c7-9c23-494a-b	1t micro	us-east-1d	8 months	Medium Utilization	\$0.88	\$0.006	1	-	active
9d5cb137-7251-4a22-8	1t micro	us-east-1d	8 months	Medium Utilization	\$1.96	\$0.006	1	-	active

### To cancel your listing

You can cancel your listing at any time, as long as it's in the **active** state. You cannot cancel the listing if it's already matched or being processed for a sale. If some of the instances in your listing are matched and you cancel the listing, only the remaining unmatched instances will be removed from the listing.

1. Go to the **Reserved Instances** page in the Amazon EC2 console, and right-click your Reserved Instance.
2. On the **My Listings** tab, click **Cancel Listing**.

The screenshot shows the 'My Listings' tab with the 'Listing State' set to 'Active'. A red arrow points from the 'Cancel Listing' button in the top right corner of the window to the 'Cancel Listing' button in the details panel.

### To change your listing

Currently, you cannot modify your listing directly. However, you can change your listing by first canceling it and then creating another listing with new parameters.

1. Cancel your active Reserved Instance listing. For more information, see the previous procedure.
2. Create a new listing. For more information, see [Listing Your Reserved Instance \(p. 251\)](#).

### Amazon EC2 CLI

#### To view your listing

- Run `ec2-describe-reserved-instances-listing` to get details about your listing.

```
PROMPT> ec2-describe-reserved-instances-listings
```

Amazon EC2 returns output similar to the following:

```
PROMPT> ec2-describe-reserved-instances-listings
Type ReservedInstancesListingId ReservedInstancesId CreateDate UpdateDate
Status StatusMessage
LISTING 615d8a10-8224-4c19-ba7d-b9aa0example 1ba8e2e3-d20d-44ec-b202-
fcb6aexample Wed Aug 22 09:02:58 PDT 2012 Wed Aug 22 14:24:26 PDT 2012 can-
celled cancelled
INSTANCE-COUNT available 0
INSTANCE-COUNT sold 0
INSTANCE-COUNT cancelled 1
INSTANCE-COUNT pending 0
PRICE-SCHEDULE 10 $1.2
PRICE-SCHEDULE 9 $1.08
PRICE-SCHEDULE 8 $0.96
PRICE-SCHEDULE 7 $0.84
PRICE-SCHEDULE 6 $0.72
PRICE-SCHEDULE 5 $0.6
PRICE-SCHEDULE 4 $0.48
PRICE-SCHEDULE 3 $0.36
PRICE-SCHEDULE 2 $0.24
PRICE-SCHEDULE 1 $0.12
LISTING d5fa5166-83c3-40e4-abb2-b7298example 1ba8e2e3-d20d-44ec-b202-
fcb6aexample Wed Aug 22 14:31:55 PDT 2012 Wed Aug 22 14:42:40 PDT 2012 closed-
closed
INSTANCE-COUNT available 0
INSTANCE-COUNT sold 1
INSTANCE-COUNT cancelled 0
INSTANCE-COUNT pending 0
PRICE-SCHEDULE 10 $0.9
PRICE-SCHEDULE 9 $0.81
PRICE-SCHEDULE 8 $0.72
PRICE-SCHEDULE 7 $0.63
PRICE-SCHEDULE 6 $0.54
PRICE-SCHEDULE 5 $0.45
PRICE-SCHEDULE 4 $0.36
PRICE-SCHEDULE 3 $0.27
PRICE-SCHEDULE 2 $0.18
PRICE-SCHEDULE 1 $0.09
....
```

```
LISTING 095c0e18-c9e6-4692-97e5-653e0example b847fa93-c736-4eae-bca1-e3147example Tue Aug 28 18:21:07 PDT 2012 Tue Aug 28 18:21:07 PDT 2012 active active
INSTANCE-COUNT available 1
INSTANCE-COUNT sold 0
INSTANCE-COUNT cancelled 0
INSTANCE-COUNT pending 0
PRICE-SCHEDULE 5 $1.2
PRICE-SCHEDULE 4 $1.2
PRICE-SCHEDULE 3 $1.2
PRICE-SCHEDULE 2 $1.2
PRICE-SCHEDULE 1 $1.2
```

### To cancel your listing

You can cancel your listing at any time, as long as it's in the *active* state. You cannot cancel the listing if it's already matched or being processed for a sale.

- Run `ec2-cancel-reserved-instances-listing` to cancel your listing.

```
PROMPT> ec2-cancel-reserved-instances-listing 095c0e18-c9e6-4692-97e5-653e0example
```

Amazon EC2 returns output similar to the following:

```
PROMPT> ec2-cancel-reserved-instances-listing
Type ReservedInstancesListingId ReservedInstancesId CreateDate UpdateDate
Status StatusMessage
LISTING 095c0e18-c9e6-4692-97e5-653e0example b847fa93-c736-4eae-bca1-e3147example Tue Aug 28 18:21:07 PDT 2012 Tue Aug 28 18:21:07 PDT 2012 cancelled cancelled
INSTANCE-COUNT available 0
INSTANCE-COUNT sold 0
INSTANCE-COUNT cancelled 1
INSTANCE-COUNT pending 0
PRICE-SCHEDULE 5 $1.2
PRICE-SCHEDULE 4 $1.2
PRICE-SCHEDULE 3 $1.2
PRICE-SCHEDULE 2 $1.2
PRICE-SCHEDULE 1 $1.2
```

### To change your listing

Currently, you cannot modify your listing directly. However, you can change your listing by first canceling it and then creating another listing with new parameters.

1. Cancel your active Reserved Instance listing. For more information, see the previous procedure.
2. Create a new listing. For more information, see [Listing Your Reserved Instance \(p. 251\)](#).

## Amazon EC2 API

### To view your listing

- Call `DescribeReservedInstancesListings` to get details about your listing.

The command should look like this example:

```
http://ec2.amazonaws.com/?Action=DescribeReservedInstancesListings
&AUTHPARAMS
```

Following is an example response.

```
<DescribeReservedInstancesListingsResponse>
 <requestId>cec5c904-8f3a-4de5-8f5a-ff7f9example</requestId>
 <reservedInstancesListingsSet>
 <item>
 <reservedInstancesListingId>5ec28771-05ff-4b9b-aa31-
9e57dexample</reservedInstancesListingId>
 <reservedInstancesId>f127bd27-cee4-443a-a76b-a5af9example</re
servedInstancesId>
 <createDate>2012-08-30T17:11:09.449Z</createDate>
 <updateDate>2012-08-30T21:00:42.300Z</updateDate>
 <status>active</status>
 <statusMessage>active</statusMessage>
 <instanceCounts>
 <item>
 <state>Available</state>
 <instanceCount>2</instanceCount>
 </item>
 <item>
 <state>Sold</state>
 <instanceCount>1</instanceCount>
 </item>
 <item>
 <state>Cancelled</state>
 <instanceCount>0</instanceCount>
 </item>
 <item>
 <state>Pending</state>
 <instanceCount>0</instanceCount>
 </item>
 </instanceCounts>
 <priceSchedules>
 <item>
 <term>11</term>
 <price>2.5</price>
 <currencyCode>USD</currencyCode>
 <active>true</active>
 </item>
 <item>
 <term>10</term>
 <price>2.5</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
```

```
<term>9</term>
<price>2.5</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>8</term>
<price>2.0</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>7</term>
<price>2.0</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>6</term>
<price>2.0</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>5</term>
<price>1.5</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>4</term>
<price>1.5</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>3</term>
<price>0.7</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>2</term>
<price>0.7</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>1</term>
<price>0.1</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
</priceSchedules>
<tagSet/>
<clientToken>listRI1</clientToken>
</item>
```

```
 </reservedInstancesListingsSet>
</DescribeReservedInstancesListingsResponse>
```

## To cancel your listing

You can cancel your listing at any time, as long as it's in the *active* state. You cannot cancel the listing if it's already matched or being processed for a sale.

- Run `CancelReservedInstancesListing` to cancel your listing `5ec28771-05ff-4b9b-aa31-9e57dexample` and remove it from the Reserved Instance Marketplace.

The command should look like this example:

```
https://ec2.amazonaws.com/?Action=CancelReservedInstancesListing
&ReservedInstancesListingId.0=5ec28771-05ff-4b9b-aa31-9e57dexample
&AUTHPARAMS
```

Following is an example response.

```
<CancelReservedInstancesListingResponse>
 <requestId>bec2cf62-98ef-434a-8a15-886fcexample</requestId>
 <reservedInstancesListingsSet>
 <item>
 <reservedInstancesListingId>5ec28771-05ff-4b9b-aa31-
9e57dexample</reservedInstancesListingId>
 <reservedInstancesId>f127bd27-cee4-443a-a76b-a5af9example</re
servedInstancesId>
 <createDate>2012-08-30T17:11:09.449Z</createDate>
 <updateDate>2012-08-31T14:12:23.468Z</updateDate>
 <status>cancelled</status>
 <statusMessage>cancelled</statusMessage>
 <instanceCounts>
 <item>
 <state>Available</state>
 <instanceCount>0</instanceCount>
 </item>
 <item>
 <state>Sold</state>
 <instanceCount>1</instanceCount>
 </item>
 <item>
 <state>Cancelled</state>
 <instanceCount>2</instanceCount>
 </item>
 <item>
 <state>Pending</state>
 <instanceCount>0</instanceCount>
 </item>
 </instanceCounts>
 <priceSchedules>
 <item>
 <term>11</term>
 <price>2.5</price>
 <currencyCode>USD</currencyCode>
 </item>
 </priceSchedules>
 </item>
 </reservedInstancesListingsSet>
</CancelReservedInstancesListingResponse>
```

```
 <active>false</active>
 </item>
 <item>
 <term>10</term>
 <price>2.5</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>9</term>
 <price>2.5</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>8</term>
 <price>2.0</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>7</term>
 <price>2.0</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>6</term>
 <price>2.0</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>5</term>
 <price>1.5</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>4</term>
 <price>1.5</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>3</term>
 <price>0.7</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>2</term>
 <price>0.7</price>
 <currencyCode>USD</currencyCode>
 <active>false</active>
 </item>
 <item>
 <term>1</term>
```

```
<price>0.1</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
</priceSchedules>
<tagSet/>
<clientToken>listRI1</clientToken>
</item>
</reservedInstancesListingsSet>
</CancelReservedInstancesListingResponse>
```

In the response to the Cancel request, you see the canceled listing. Another way to see that canceled listing is by calling `DescribeReservedInstancesListings`. The request will look like this example:

```
http://ec2.amazonaws.com/?Action=DescribeReservedInstancesListings
&AUTHPARAMS
```

The response will look like the following example:

```
<DescribeReservedInstancesListingsResponse>
<requestId>bec2cf62-98ef-434a-8a15-367c0example</requestId>
<reservedInstancesListingsSet>
<item>
<reservedInstancesListingId>5ec28771-05ff-4b9b-aa31-
9e57dexample</reservedInstancesListingId>
<reservedInstancesId>f127bd27-cee4-443a-a76b-a5af9example</re-
servedInstancesId>
<createDate>2012-08-30T17:11:09.449Z</createDate>
<updateDate>2012-08-31T14:12:23.468Z</updateDate>
<status>cancelled</status>
<statusMessage>cancelled</statusMessage>
<instanceCounts>
<item>
<state>Available</state>
<instanceCount>0</instanceCount>
</item>
<item>
<state>Sold</state>
<instanceCount>1</instanceCount>
</item>
<item>
<state>Cancelled</state>
<instanceCount>2</instanceCount>
</item>
<item>
<state>Pending</state>
<instanceCount>0</instanceCount>
</item>
</instanceCounts>
<priceSchedules>
<item>
<term>11</term>
<price>2.5</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
```

```
<item>
<term>10</term>
<price>2.5</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>9</term>
<price>2.5</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>8</term>
<price>2.0</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>7</term>
<price>2.0</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>6</term>
<price>2.0</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>5</term>
<price>1.5</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>4</term>
<price>1.5</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>3</term>
<price>0.7</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>2</term>
<price>0.7</price>
<currencyCode>USD</currencyCode>
<active>false</active>
</item>
<item>
<term>1</term>
<price>0.1</price>
<currencyCode>USD</currencyCode>
```

```
 <active>false</active>
 </item>
</priceSchedules>
<tagSet/>
<clientToken>listRI1</clientToken>
</item>
</reservedInstancesListingsSet>
</DescribeReservedInstancesListingsResponse>
```

## To change your listing

Currently, you cannot modify your listing directly. However, you can change your listing by first canceling it and then creating another listing with new parameters.

1. Cancel your active Reserved Instance listing. For more information, see the previous procedure.
2. Create a new listing. For more information, see [Listing Your Reserved Instance \(p. 251\)](#).

## After Your Reserved Instance Is Sold

### Topics

- [Reserved Instance Listing States \(p. 270\)](#)
- [Lifecycle of a Listing \(p. 271\)](#)
- [Getting Paid \(p. 271\)](#)
- [Notifications \(p. 272\)](#)

When your Reserved Instance is sold, AWS will send you an email notification. Each day that there is any kind of activity (for example, you create a listing; you sell a listing; or AWS sends funds to your account), you will get one email notification capturing all the activities of the day. For more information, see [Notifications \(p. 272\)](#).

You can track the status of your Reserved Instance listings by looking at the **My Listings** tab of the selected Reserved Instance on the **Reserved Instance** page in the Amazon EC2 console. The tab contains the **Listing State** as well as information about the term, listing price, and a breakdown of how many instances in the listing are available, pending, sold, and cancelled. You can also use the `ec2-describe-reserved-instances-listings` CLI command or the `DescribeReservedInstancesListings` API call, with the appropriate filter to obtain information about your Reserved Instance listings.

## Reserved Instance Listing States

**Listing State** displays the current status of your Reserved Instance listings:

- **Active**—The listing is available for purchase.
- **Cancelled**—The listing is canceled and won't be available for purchase in the marketplace.
- **Closed**—The Reserved Instance is not listed. A Reserved Instance might be *Closed* because the sale of the listing was completed.

The information displayed by **Listing State** is about the status of your listing in the Reserved Instance Marketplace. It is different from the status information that is displayed by the **State** column in the Reserved

Instance page. This **State** information is about your Reserved Instance. For more information, see [Reserved Instance States \(p. 237\)](#).

## Lifecycle of a Listing

Now that you have created a listing, let's walk through what happens when your listing sells.

When *all* the instances in your listing are matched and sold, the **My Listings** tab shows that your **Total instance count** matches the count listed under **Sold**, there are no **Available** instances left for your listing, and its **Status** is *closed*.

When only *a portion* of your listing is sold, AWS retires the Reserved Instances in the listing and creates the number of Reserved Instances equal to the Reserved Instances remaining in the count. So, the Reserved Instances listing ID and the listing that it represents, which now has an instance count of fewer instances for sale, is still active.

Any future sales of Reserved Instances in this listing are processed this way. When all the Reserved Instances in the listing are sold, AWS marks the listing as *closed*.

For example, let's say you created a listing *Reserved Instances listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample* with an instance count of 5.

Your **My Listings** tab in the **Reserved Instance** page of the Amazon EC2 console will display the listing this way:

*Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample*

- Total instance count = 5
- Sold = 0
- Available = 5
- Status = active

Let's say that a buyer purchases two of the instances, which leaves a count of three instances still available for sale. As a result of this partial sale, AWS creates a new Reserved Instance with an instance count of three to represent the remaining three that are still for sale.

This is how your listing will look in your **My Listings** tab:

*Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample*

- Total instance count = 5
- Sold = 2
- Available = 3
- Status = active

If you decide to cancel your listing and a portion of that listing has already sold, the cancellation is not effective on the portion that has been sold. Only the portion of the listing not yet sold will no longer be available in the Reserved Instance Marketplace.

## Getting Paid

As soon as AWS receives funds from the buyer of your Reserved Instance, AWS sends a message to your email address—that is, the email address associated with the account that is registered as owner of the Reserved Instance that was sold.

AWS sends an Automated Clearing House (ACH) wire transfer to the bank account that you specified when you registered for the Reserved Instance Marketplace. Typically, this transfer occurs between one

to three days after your Reserved Instance has been matched. You can view the state of this disbursement by viewing your Reserved Instance disbursement report. Disbursements take place once a day. Keep in mind that you will not be able to receive disbursements until AWS has received verification from your bank. This period can take up to two weeks.

The Reserved Instance you sold will continue to appear in the results of `DescribeReservedInstances` calls you make for another 60 days before disappearing from the list. You will receive a pro-rated refund for the portion of the upfront fee you paid for the Reserved Instance that you did not use.

## Notifications

As a seller in the Reserved Instance Marketplace, you will receive an email digest of the Reserved Instance Marketplace activities pertaining to your account. On any given day, you will receive one email digest, and you will only receive this email if one or a combination of activities occurred that day:

- You created a new listing in the Reserved Instance Marketplace.
- You sold one or more of the Reserved Instances you listed.
- AWS posted a disbursement to your bank account as a result of a sale of part or all of your listing in the Reserved Instance Marketplace.

Your email digest will look similar to the following:



Dear John Doe,

Your account's Reserved Instance Marketplace activity for 2012/08/27 is shown below:

**New Listings**

Platform	Instance Type	Offering Type	Term	Availability Zone	Price	Count	Total Potential Earnings
Linux/UNIX	m3Large	Medium Utilization	2 months	us-east-1a	\$24.44	2	\$48.88

**Recent Sales**

Platform	Instance Type	Offering Type	Term	Availability Zone	Price	Count	Total Earnings
Linux/UNIX (Amazon VPC)	m3Large	Heavy Utilization	35 months	us-east-1a	\$1,489.50	1	\$1,489.50
Linux/UNIX (Amazon VPC)	m3Large	Medium Utilization	35 months	us-east-1b	\$1,489.50	1	\$1,489.50

**Disbursements**

Zip Code	Country	Disbursement Amount	Status
12065	United States	\$1,056.11	Success
	Total	\$1,056.11	

We initiated a transfer to your checking account (ending with 2634) of \$1,056.11 on 2012/07/19. Funds usually arrive within 3 - 5 banking days, but times vary by bank.

If you have questions about your Reserved Instance Marketplace activity above, please visit the Reserved Instance Marketplace web page at <http://aws.amazon.com/reserved-instance-marketplace>.

Sincerely,  
The Amazon EC2 Team

## Requirements Checklist for Reserved Instances

Amazon EC2 Reserved Instances and the Reserved Instance Marketplace can be a powerful and cost-saving strategy for running your business. However, before you use Reserved Instances or the Reserved Instance Marketplace, ensure that you meet the requirements for purchase and sale. You also must understand the details and restrictions on certain elements of Reserved Instances and the Reserved Instance Marketplace—including seller registration, banking, using the AWS Free tier, dealing with cancelled instances, and so on. Use this topic as a checklist for buying and selling Reserved Instances, and for buying and selling in the Reserved Instance Marketplace.

### Reserved Instances

- **AWS account**—You need to have an AWS account in order to purchase Reserved Instances. If you don't have an AWS account, you should read and complete the instructions described in [Getting Started with Amazon EC2 Linux Instances \(p. 8\)](#), which provides information on signing up for your Amazon EC2 account and credentials.
- **AWS free tier**—The AWS free usage tier is available for new AWS accounts. If you are using the AWS free usage tier to run Amazon EC2 instances, and then you purchase a Reserved Instance, you will be charged for the Reserved Instance under standard pricing guidelines. For information about the free tier and the applicable services and usage amounts, see [AWS Free Usage Tier](#).

### **Buying Reserved Instances**

- **Usage fee for Heavy Utilization**—With Light Utilization and Medium Utilization Reserved Instances, you pay a one-time upfront fee and then only pay the hourly price when you use the instance. With Heavy Utilization Reserved Instances, you pay a low, one-time upfront fee and commit to paying an hourly rate for every hour of the Reserved Instance's term *whether or not you use it*. For information, see [Choosing Reserved Instances Based on Your Usage Plans \(p. 217\)](#).
- **Tiered discounts on purchases**—The Reserved Instance pricing tier discounts only apply to purchases made from AWS. These discounts do not apply to purchases of third-party Reserved Instances. For information, see [Understanding Discount Pricing Tiers \(p. 229\)](#).
- **Cancelation of purchase**—Before you confirm your purchase, review the details of the Reserved Instances that you plan to buy, and make sure that all the parameters are accurate. After you purchase a Reserved Instance (either from a third-party seller in the Reserved Instance Marketplace or from AWS), you cannot cancel your purchase. However, you can sell the Reserved Instance if your needs change. For information about selling your Reserved Instance in the Reserved Instance Marketplace, see [Selling Your Reserved Instances \(p. 250\)](#).

### **Selling Reserved Instances and the Reserved Instance Marketplace**

- **Seller requirement**—To become a seller in the Reserved Instance Marketplace, you must register as a seller. For information, see [Selling Your Reserved Instances \(p. 250\)](#).
- **Bank requirement**—AWS must have your bank information in order to disburse funds collected when you sell your Reserved Instance. The bank you specify must have a US address. For more information, see [Your Bank \(p. 247\)](#).
- **Tax requirement**—Sellers who have 200 or more transactions or who plan to sell \$20,000 or more in Reserved Instances will have to provide additional information about their business for tax reasons. For information, see [Tax Information \(p. 248\)](#).
- **Minimum selling**—The minimum price allowed in the Reserved Instance Marketplace is \$1.01.
- **When Reserved Instances can be sold**—Reserved Instances can be sold only after AWS has received the upfront payment and the Reserved Instance has been active (you've owned it) for at least 30 days. In addition, there must be at least a month remaining in the term of the Reserved Instance you are listing.
- **Modifying your listing**—Currently, you cannot modify your listing in the Reserved Instance Marketplace directly. However, you can change your listing by first canceling it and then creating another listing with new parameters. For information, see [Canceling and Changing Your Listings \(p. 260\)](#).
- **Selling discounted Reserved Instances**—Amazon EC2 Reserved Instances purchased at a reduced cost resulting from a tiering discount cannot be sold in the Reserved Instance Marketplace. For more information about the Reserved Instance Marketplace, see [Reserved Instance Marketplace \(p. 243\)](#).
- **Service fee**—AWS will charge you a service fee of 12 percent of the total upfront price of each Reserved Instance you sell in the marketplace. (The upfront price is the price the seller is charging for his Reserved Instance.)
- **Other AWS Reserved Instances**—Only Amazon EC2 Reserved Instances can be sold in the Reserved Instance Marketplace. Other AWS Reserved Instances, such as Amazon Relational Database Service (Amazon RDS) and Amazon ElastiCache Reserved Instances cannot be sold on the Reserved Instance Marketplace.

# Launching EC2 Instances

This section describes how to launch EC2 instances. All EC2 instances are launched from an AMI. The AMI specifies the operating system, application server, and applications that the instance will have.

All instances require a key pair so that login information can be securely communicated. You can either use an existing key pair, or create a new key pair for your instance. For more information about creating key pairs, see [Getting a Key Pair \(p. 274\)](#).

If you have a snapshot of the root device volume of a Linux/UNIX instance, you can create and launch an AMI from that snapshot. For more information about creating and launching an instance from a snapshot, see [Creating and Launching an AMI from a Snapshot \(p. 290\)](#).

## Topics

- [Getting a Key Pair \(p. 274\)](#)
- [Launching an Instance from an AMI \(p. 284\)](#)
- [Creating and Launching an AMI from a Snapshot \(p. 290\)](#)

## Getting a Key Pair

AWS uses public–key cryptography to encrypt and decrypt login information. Public–key cryptography uses a public key to encrypt a piece of data, such as a password, then the recipient uses the private key to decrypt the data. The public and private keys are known as a *key pair*. AWS only stores the public key, and you store the private key. Anyone who possesses your private key can decrypt your login information, which is why it is important to store your private keys in a secure place.

Linux instances have no password, and you use a key pair to log in to them. Likewise, with Windows instances, you use a key pair to obtain the administrator password. The public key is embedded in your instance, enabling you to use the private key to log in securely without a password. After you create your own AMIs, you can choose other mechanisms to log in securely to your new instances.

You can have multiple key pairs, and each key pair requires a name. Be sure to choose a name that is easy to remember.

You have two options for creating a key pair:

- [Have Amazon EC2 generate it for you \(p. 275\)](#). This is the preferred method, as it is the simplest. You can use the Amazon EC2 console, the `ec2-create-keypair` command, or the `CreateKeyPair` action.
- [Generate it yourself \(p. 279\)](#).

You can use a third-party tool such as OpenSSH, and then import the public key to Amazon EC2 using either the `ec2-import-keypair` command or the `ImportKeyPair` action.

### Note

In Amazon EC2, all key pairs, whether created or imported, are specific to the region that they are created in or imported to. Make sure your Amazon EC2 endpoint is correctly set before creating or importing the key pair.

Amazon EC2 doesn't store a copy of the private key for either option. Amazon EC2 only stores the public key, and associates it with a friendly name that you specify for the key pair.

**Note**

If you are using PuTTY in Windows, you must convert the private key to PuTTY's format. For more information about using PuTTY with Amazon EC2, see [Connecting to Linux/UNIX Instances from Windows Using PuTTY \(p. 301\)](#).

**Topics**

- [How to Have AWS Create the Key Pair for You \(p. 275\)](#)
- [How to Generate Your Own Key and Import It to Amazon EC2 \(p. 279\)](#)
- [Verifying Your Key Pair \(p. 283\)](#)

## How to Have AWS Create the Key Pair for You

**Topics**

- [Amazon EC2 Console \(p. 275\)](#)
- [Command Line Tools \(p. 276\)](#)
- [API \(p. 278\)](#)

### Amazon EC2 Console

#### To generate a key pair

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

Use the email address and password that you specified when signing up for AWS.

2. From the navigation bar, select the region for the key pair. This choice is important because key pair resources cannot be shared between regions. For example, if you create a key pair in the US West (Oregon) Region, you won't be able to see or use the key pair in another region.



3. Click **Key Pairs** in the navigation pane to display the list of key pairs associated with your account in the current region.



4. Click **Create Key Pair**.

**Create Key Pair**

5. Enter a name for the new key pair in the **Key Pair Name** field of the **Create Key Pair** dialog box and click **Create**.



6. The private key file will automatically be downloaded by your browser. The file name will be the key pair name with the .pem extension. In this example, the private key file will be gsg-keypair.pem. Save the private key file in a safe place. You will need it to access any instances that you launch with this key pair.

## Command Line Tools

Use the [ec2-create-keypair](#) command.

### To generate a key pair

1. The following command creates a key pair with the name gsg-keypair.

```
PROMPT> ec2-create-keypair gsg-keypair
```

Amazon EC2 returns the contents of the private key, similar to this one.

```
KEYPAIR gsg-keypair
1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f
---- BEGIN RSA PRIVATE KEY ----
MIICiTC...
VVMx...
b24x...
Bgkq...
KEYPAIR gsg-keypair
```

```

MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTA1dBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBqNVBAstC01BTSDb25z
b2x1MRIwEAYDVQQDEwlUZXN0Q2lsYWmxHzAdBgkqhkiG9w0BCQEWEg5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJaoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAArHndlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJ1lJ00zbhNYS5f6GuoEDmFJl0ZxBhjJnyp3780D8uTs7fLvjx79LjSTb
NYiytVbzPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
```

-----END RSA PRIVATE KEY-----

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

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

3. Confirm that the file contents looks similar to the following and save the file. You can save the file in any directory, but if you do not put it in your current directory, you must specify the full path when using commands that require the key pair.

```
----- BEGIN RSA PRIVATE KEY -----
MIICitCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTA1dBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBqNVBAstC01BTSDb25zb2x1MRIwEAYDVQQDEwlUZXN0Q2lsYWmxHzAd
BgkqhkiG9w0BCQEWEg5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTA1dBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBqNVBAstC01BTSDb25z
b2x1MRIwEAYDVQQDEwlUZXN0Q2lsYWmxHzAdBgkqhkiG9w0BCQEWEg5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJaoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAArHndlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJ1lJ00zbhNYS5f6GuoEDmFJl0ZxBhjJnyp3780D8uTs7fLvjx79LjSTb
NYiytVbzPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
```

-----END RSA PRIVATE KEY-----

4. If you're using OpenSSH (or another SSH client), you should set the permissions of the private key file so it is only readable by you.

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

```
$ chmod 400 gsg-keypair.pem ; ls -l gsg-keypair.pem
```

You receive output similar to the following example.

```
-r----- 1 Jane Doe 1701 Jun 19 17:57 gsg-keypair.pem
```

5. Verify that the key pair is valid as shown in [Verifying Your Key Pair \(p. 283\)](#).

## API

Use the [CreateKeyPair](#) action.

### To generate a key pair

1. Construct the following Query request.

```
https://ec2.amazonaws.com/
?Action=CreateKeyPair
&KeyName=gsg-keypair
&AUTHPARAMS
```

The following is an example response.

```
<CreateKeyPairResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
 <keyName>gsg-keypair</keyName>
 <keyFingerprint>
 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f
 </keyFingerprint>
 <keyMaterial>---- BEGIN RSA PRIVATE KEY ----
MIICITCCAFICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxszAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC01BTSDb25zb2x1MRIwEAYDVQQDEw1UZXN0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEg5vb25lQGftYXpbvi5jb20wHhcNMTEwNDI1Mja0NTIxWhcN
MTIwNDI0Mja0NTIxWjCBiDELMAkGA1UEBhMCVVMxszAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC01BTSDb25z
b2x1MRIwEAYDVQQDEw1UZXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEg5vb25lQGft
YXpbvi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTc2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJ1lJ00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs7fLvjx79LjSTb
NYiytVbZPQU5Yaxu2jXnimvw3rrszlaEXAMPLE=
 -----END RSA PRIVATE KEY-----</keyMaterial>
</CreateKeyPairResponse>
```

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

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

3. Confirm that the file contents looks similar to the following and save the file. You can save the file in any directory, but if you do not put it in your current directory, you must specify the full path when using commands that require the key pair.

```
----- BEGIN RSA PRIVATE KEY -----
MIICITCCAFICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxszAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC01BTSDb25zb2x1MRIwEAYDVQQDEw1UZXN0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEg5vb25lQGftYXpbvi5jb20wHhcNMTEwNDI1Mja0NTIxWhcN
MTIwNDI0Mja0NTIxWjCBiDELMAkGA1UEBhMCVVMxszAJBgNVBAgTAldBMRAwDgYD
```

```
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAstC01BTSBDb25z
b2x1MRIwEAYDVQQDEwlUZXN0Q2lsYWmxHzAdBgkqhkiG9w0BCQEWEg5vb25lQGFT
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END RSA PRIVATE KEY-----
```

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

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

```
$ chmod 400 gsg-keypair.pem ; ls -l gsg-keypair.pem
```

You receive output similar to the following example.

```
-r----- 1 Jane Doe 1701 Jun 19 17:57 gsg-keypair.pem
```

5. Verify that the key pair is valid as shown in [Verifying Your Key Pair \(p. 283\)](#).

## How to Generate Your Own Key and Import It to Amazon EC2

This section describes how to import a public key to Amazon EC2 from a key pair you've created with a third-party tool.

You can create an RSA key pair on Windows or Linux using the `ssh-keygen` command line tool (provided with the standard OpenSSH installation). Java, Ruby, Python, and many other programming languages provide standard libraries for RSA key pair creation.

Amazon EC2 accepts the following formats:

- OpenSSH public key format (e.g., the format in `~/.ssh/authorized_keys`)
- Base64 encoded DER format
- SSH public key file format as specified in [RFC4716](#)

Amazon EC2 does not accept DSA keys. Make sure your key generator is set up to create RSA keys.

Supported lengths: 1024, 2048, and 4096.

### Generate the key pair

#### To generate the key pair to be imported

1. Generate the key pair with a third-party tool of your choice.
2. Save the public key to a local file. The extension for this file is not important. In this example, this filename is `C:\keys\gsg-keypair.ppk`.

3. Save the private key to a different local file that has the .pem extension. In this example, this filename is C:\keys\gsg-keypair.pem. Save the private key file in a safe place. You will need it to access any instances that you launch with this key pair.

## Import the Key Pair

### Topics

- [Amazon EC2 Console \(p. 280\)](#)
- [Command Line Tools \(p. 281\)](#)
- [API \(p. 282\)](#)

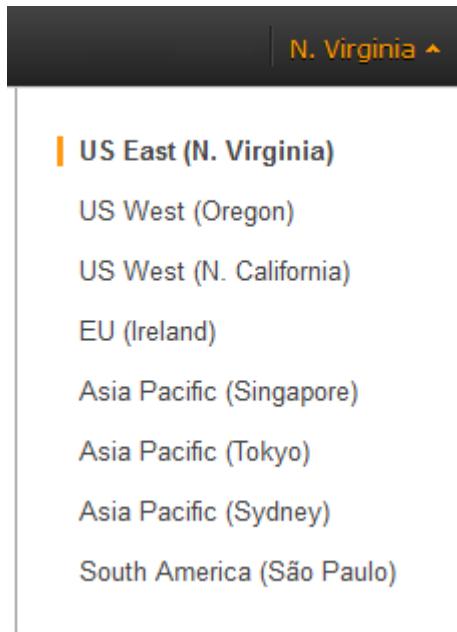
### Amazon EC2 Console

#### To import a public key

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

Use the email address and password that you specified when signing up for AWS.

2. From the navigation bar, select the region for the key pair. This choice is important because key pair resources cannot be shared between regions. For example, if you import a key pair into the US West (Oregon) Region, you won't be able to see or use the key pair in another region.



3. Click **Key Pairs** in the navigation pane to display the list of key pairs associated with your account in the current region.



4. Click **Import Key Pair**.

**Import Key Pair**

5. In the **Import Keypair** dialog, select the public key file saved above, enter a name for the key pair in the **Keypair Name** field, and click **Yes, Import**.



6. After the public key file is imported, verify that the key pair is valid as shown in [Verifying Your Key Pair \(p. 283\)](#).

## Command Line Tools

### To import a public key

1. Use the [ec2-import-keypair](#) command to import the public key file to Amazon EC2. The following example names the key pair `gsg-keypair`. The response displays the MD5 public key fingerprint as specified in section 4 of [RFC4716](#).

- After the public key file is imported, verify that the key pair is valid as shown in [Verifying Your Key Pair \(p. 283\)](#).

API

### To import the public key

1. Use the [ImportKeyPair](#) action to import the public key file to Amazon EC2. The following Query example names the key pair `gsg-keypair`. You must base64 encode the public key material before sending it to Amazon EC2.

```
https://ec2.amazonaws.com/?Action=ImportKeyPair
&KeyName=gsg-keypair
&PublicKeyMaterial=LS0tLS1CRUdJTIBDRVJUSUZJQ0FURS0tLS0tDQpNSULDZHpdQ0F1Q2dBd01CQWdJR0FQa1RyR3pQ
TUEwR0NTcUdTSWIzRFFFQkJRVUFNRk14Q3pBSkJnTlZCQV1UDQpBbFZUTVJnd0VRWURWUVFLRxwQmJXRjzImjR1WTI5dE1Rd3dZ11EV1FRTEV3TkJWMU14SVRBZkJnTlZCQV1UDQpHRUZYVX1CTWFxmXBkR1ZrTFVGemMzVn1ZVzVqWlNCRFFUQWVGdzB3T1RBM016RxlNVFEzTxpWYUZ3MhHnREezDQpNekv5TVRRM016VmFNRk14Q3pBSkJnTlZCQV1UQWxWVE1STXdFUV1EV1FRS0V3cEJiV0Y2Yj10dVkyOXRNUnM3DQpGUV1EV1FRTEV3NUJWMU10UkdWM1pXeHZjR1Z5Y3pFVk1CTUdBMVVFQXhNTWJUSnViRGhxZW00MWVHUjFNSUdmDQpNQTBHQ1NxR1NJYjNEUUVQCvFVQUE0R05BRENcavFLQmdRQ1d0azBoQytrcExBRnp2YkFQc3U1TDU5bFMwUnI0DQprZEpaM0RFak1pL0IwV2ZDSzhpS2hWYwt1WithHSnJtNDdMUHZCaFVKWk91eHVUU0VXakFDNmlybDJzKz1SWxVjDQpFZXg0Tj14Z1pCZGpOR1AzdEgwZ2NuWjd1bXZ4aFBrtETt0RTdpZmViNmNGWUhRdHpHRnRPQ0ZQtmdUSE92VDE5DQoyR31zb1Vyu3BDVGFCUU1EQVFBQm8xY3dWVFpQmdoVkhROEBZjhFQkFNQ0JhQXdgz11EV1IwbEFRSC9CQxd3DQpDZ11JS3dzQkJRVUhBd013REFZRFZSMFRBUUgvQkFJd0FEQWRcz05WSFE0RUznUVU1RVNuTUZZUzdyTDNxTUDLDQpqejmxDVXZ5TTThnMHDcUV1KS29aSwH2Y05BUUVGQ1FBRGdZRUFnWjdDz11JWHR1WFm1NHVqbU5jOTR0NWRNc3krDQpCM0Z3WVVNdud4WUI2eGQySUVWMTFLRVEyZ0hpZUdMu21juWg4c2JXTTdtKzcrYm9UNmc2U2hLBu1jb1kzWkRTDQpWRVFZ225qcEt1aEZRd2pmaVpTUEc1UG5SvENhdkvqs31TTUpDVGxpdtDdTtjmR2J3cFU5Uzg3K21GM2tsMGrmDQpZN1IrbE15SWcrU3ROOTg9DQotLS0tLUVRCBDRVJUSUZJQ0FURS0tLS0tEXAMPLE
&AuthParams
```

The response includes the MD5 public key fingerprint as specified in section 4 of [RFC4716](#).

2. After the public key file is imported, verify that the key pair is valid as shown in [Verifying Your Key Pair \(p. 283\)](#).

## Verifying Your Key Pair

To verify that the key pair you have created works properly, you verify that the new key pair exists in Amazon EC2, and then verify that the new key pair works with an instance.

### Topics

- [Verify the Existence of the Key Pair \(p. 283\)](#)
- [Launch a New Instance \(p. 284\)](#)

## Verify the Existence of the Key Pair

You can verify that your key pair has been created or imported correctly in one of the following ways. Remember that key pairs are specific to regions, so if the currently selected region is not the same as the region that the key pair was created in or imported to, the new key pair will not be found.

### Topics

- [Amazon EC2 Console \(p. 283\)](#)
- [Command Line Tools \(p. 284\)](#)
- [API \(p. 284\)](#)

### Amazon EC2 Console

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

Use the email address and password that you specified when signing up for AWS.

2. From the navigation bar, select the region to list the key pairs for. This choice is important because key pair resources cannot be shared between regions. For example, if you import a key pair into the US West (Oregon) Region, you won't be able to see or use the key pair in another region.



3. Click **Key Pairs** in the navigation pane.



4. Verify that the key pair that was created or imported exists in the list of key pairs associated with your account in the current region.

	Key Pair Name	Fingerprint
<input type="checkbox"/>	gsg-keypair	1e:39:93:b0:99:21:32:56:05:c0:30:04:63:48:9f
<input type="checkbox"/>	keypair-1	7e:ba:01:6f:79:3f:72:a0:58:92:d3:3d:a8:10:35:65:14:7e:1d:a3
<input type="checkbox"/>	keypair-2	1b:ee:50:03:1a:72:60:a0:65:a6:74:98:69:a6:1d:a0:a6:45:00:4e

## Command Line Tools

Use the [ec2-describe-keypairs](#) command to list the key pairs currently associated with your account.

## API

Use the [DescribeKeyPairs](#) action to list the key pairs currently associated with your account.

## Launch a New Instance

To verify that the key pair works correctly, launch a new Linux instance that uses the new key pair, and then connect to that instance. For more information about launching Linux instances, see [Step 2: Launch an Amazon EC2 Instance \(p. 9\)](#). In step 6 of the procedure, under **Choose a Key Pair**, select the **Select Existing** option, and select the key pair you created from the drop-down list.

After the instance is running, connect to it as shown in [Step 3: Connect to Your Instance \(p. 13\)](#). In step 8 of the procedure, enter the path and file name of the private key file you saved. In this example, the private key file name is `gsg-keypair.pem`. If you can successfully connect to the instance, the key pair is working correctly.

# Launching an Instance from an AMI

This topic describes how to launch and run an instance from an AMI.

**Important**

After launching an instance, you are billed hourly for running time. When you are finished with an instance, be sure to terminate it. For more information, see [Terminating Instances \(p. 380\)](#).

If the instance's state immediately goes to *terminated* instead of *running*, you can get information about why the instance didn't launch. For more information, see [What To Do If An Instance Immediately Terminates \(p. 412\)](#).

**Topics**

- [AWS Management Console \(p. 285\)](#)
- [Command Line Tools \(p. 287\)](#)
- [API \(p. 288\)](#)

## AWS Management Console

You can either leverage the Free Usage Tier to launch and use a free Amazon EC2 Micro instance for 12 months, or you can launch a larger instance type, but not within the Free Usage Tier. For more information, see the [AWS Free Usage Tier product page](#) and [Getting Started with AWS Free Usage Tier](#).

If you launch an instance that is not within the Free Usage Tier, you incur the standard Amazon EC2 usage fees for the instance. For more information about Amazon EC2 usage rates, see the [Amazon EC2 product page](#).

**Important**

If you launch an instance that is not within the Free Usage Tier, you are billed after you launch the instance and charged for the time that the instance is running, even if it remains idle.

### To launch an instance

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region for the instance. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. Select the region that meets your needs. For more information, see [Resource Locations \(p. 534\)](#).



3. From the Amazon EC2 console dashboard, click **Launch Instance**.

The **Create a New Instance** page provides the following ways to launch an instance:

- The **Classic Wizard** provides more granular control and advanced settings for configuring your instance.
- The **Quick Launch Wizard** simplifies the process for you by automatically configuring many selections to get you started quickly with your instance. We recommend this wizard if you are just trying out Amazon EC2. For more information, see [Getting Started with Amazon EC2 Linux Instances \(p. 8\)](#).
- The **AWS Marketplace** is an online store where you can buy software that runs on AWS. You can launch your instance with just one click.

For this procedure, use the Classic Wizard. Click **Classic Wizard**, and then click **Continue**.

4. On the **CHOOSE AN AMI** page, the **Quick Start** tab displays a list of basic configurations called Amazon Machine Images (AMIs). An AMI contains all the information needed to create a new instance. For example, an AMI might contain the software required to act as a web server (for example, Linux, Apache, and your website).

To keep things simple, AWS marks the AMIs that are available in the Free Usage Tier with a star.

As you are selecting an AMI, it's important to note whether the AMI is backed by an instance store or by Amazon EBS. Select the type of AMI that meets your needs. For more information, see [Storage for the Root Device \(p. 19\)](#).

Choose the AMI to use and click **Select**.

5. On the **INSTANCE DETAILS** page, change the following settings as necessary, and then click **Continue**.

- **Number of Instances:** Select the number of instances based on the selected AMI to launch.
- **Instance Type:** Select the size of the instance to launch. Larger instance types have more CPU and memory.

To stay within the Free Usage Tier, choose `Micro (t1.micro)`.

- **Launch as an EBS-Optimized Instance:** If the instance type supports this feature, select this check box to enable it. (Otherwise, this option is grayed out.) Additional charges apply. For more information, see [EBS-Optimized Instances \(p. 91\)](#).
- **Availability Zone:** Select the Availability Zone to use. We recommend that you choose `No Preference`.

6. On the second **INSTANCE DETAILS** page, change the following advanced instance options as necessary, and then click **Continue**.

- **Kernel ID:** Select `Use Default` unless you want to use a specific kernel. For more information, see [Kernel FAQ \(p. 84\)](#).
- **RAM Disk ID:** Select `Use Default` unless you want to use a specific RAM disk. For more information, see [Kernel FAQ \(p. 84\)](#).
- **Monitoring:** Select this check box to enable detailed monitoring of your instance using Amazon CloudWatch. Additional charges apply. For more information, see [Introduction to Amazon CloudWatch](#) in the *Amazon CloudWatch Developer Guide*.
- **User Data:** You can specify user data to configure an instance during launch or to run a configuration script.
- **base64 encoded:** Select this check box if you want data to be base64 encoded.

- **Termination Protection:** Select this check box to prevent against accidental termination. For more information, see [Enabling Termination Protection for an Instance \(p. 381\)](#).
  - **Shutdown Behavior:** Select whether the instance should stop or terminate when shut down.
  - **IAM Role:** Select an AWS Identity and Access Management (IAM) role to associate with the instance. For more information, see [IAM Roles for Amazon EC2 \(p. 425\)](#).
7. On the third **INSTANCE DETAILS** page, you can click **Edit** to specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume), and then click **Continue**. For this example, it's fine to stick with the default root volume. For more information about changing the block device mapping, see [Updating the Block Device Mapping when Launching an Instance \(p. 528\)](#).
  8. On the final **INSTANCE DETAILS** page, specify tags for the instance, and then click **Continue**. For more information about tags, see [Tagging Your Amazon EC2 Resources \(p. 540\)](#).
  9. On the **CREATE KEY PAIR** page, you can choose from any existing key pairs that you have created in the current region, or create a new one. For this example, click **Choose from your existing Key Pairs** and select `gsg-keypair` from the list of existing key pairs. For more information about creating key pairs, see [Getting a Key Pair \(p. 274\)](#).

**Important**

Do not select the **Proceed without a Key Pair** option. If you launch an instance without a key pair, you can't connect to it. This option is used only when you are creating your own AMI and don't need to connect to the instance.

10. On the **CONFIGURE FIREWALL** page, the wizard automatically selects a security group for you.

A security group defines firewall rules for your instances. These rules specify which incoming network traffic is delivered to your instance. All other traffic is ignored.

If you're new to Amazon EC2 and haven't set up any security groups yet, AWS defines a default security group for you. If you use an AMI on the **Quick Start** tab, the name and description for the group is `quick-start-x` where `x` is a number associated with your quick-start group. The first security group you create using the **Quick Start** tab is named `quick-start-1`. You can change the name and description. For an Amazon Linux instance, you connect through SSH on port 22. The `quick-start-x` security group automatically allows SSH traffic on port 22. For a Windows instance, you connect through Remote Desktop Protocol (RDP) and the security group automatically allows RDP traffic on port 3389.

If you have used Amazon EC2 before, the wizard looks for an existing security group for the type of instance you're creating.

**Caution**

The `quick-start-x` security group enables all IP addresses to access your instance over the specified ports (that is, SSH for Linux/UNIX or RDP for Windows). This is acceptable for the short exercise in this tutorial, but it's unsafe for production environments. In production, you authorize only a specific IP address or range of addresses to access your instance.

**Note**

After an instance is running, you can't change which security groups it belongs to, but you can add and remove rules.

11. Click **Continue** to review your settings. When you're satisfied with your selections, click **Launch** to begin launching your instance.

## Command Line Tools

### To launch an instance

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

```
PROMPT> ec2-run-instances ami-1a2b3c4d -k gsg-keypair
```

Amazon EC2 returns output similar to the following example.

```
RESERVATION r-1a2b3c4d 111122223333 default
INSTANCE i-1a2b3c4d ami-1a2b3c4d pending gsg-keypair 0 m1.small YYYY-MM-
DDTHH:MM:SS+0000 us-east-1a aki-1a2b3c4d monitoring-disabled ebs
paravirtual xen sg-1a2b3c4d default false
```

2. Look for the instance ID in the second field and write it down.

You use it to manipulate this instance (including terminating it when you are finished). It takes a few minutes for the instance to launch.

3. Use the [ec2-describe-instances](#) command to display the launch status of the instance.

```
PROMPT> ec2-describe-instances i-1a2b3c4d
RESERVATION r-1a2b3c4d 111122223333 default
INSTANCE i-1a2b3c4d ami-1a2b3c4d ec2-67-202-51-223.compute-1.amazonaws.com
ip-10-251-50-35.ec2.internal running gsg-keypair 0 m1.small YYYY-MM-
DDTHH:MM:SS+0000 us-east-1a aki-1a2b3c4d monitoring-disabled 184.73.10.99
10.254.170.223 ebs paravirtual xen ABCDE1234567890123 sg-1a2b3c4d
default false
BLOCKDEVICE /dev/sda1 vol-1a2b3c4d YYYY-MM-DDTHH:MM:SS.SSSZ
true
```

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

## API

### To launch an instance

1. Use the [RunInstances](#) action. Construct the following request, where `ami_id` represents the latest version of an AMI.

```
https://ec2.amazonaws.com/
?Action=RunInstances
&ImageId=ami-1a2b3c4d
&MaxCount=1
&MinCount=1
&KeyName=gsg-keypair
&Placement.AvailabilityZone=us-east-1a
&AUTHPARAMS
```

The following is an example response.

```
<RunInstancesResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
 <reservationId>r-1a2b3c4d</reservationId>
 <ownerId>111122223333</ownerId>
 <groupSet>
 <item>
 <groupId>default</groupId>
 </item>
 </groupSet>
 <instancesSet>
 <item>
 <instanceId>i-1a2b3c4d</instanceId>
 <imageId>ami-1a2b3c4d</imageId>
 <instanceState>
 <code>0</code>
 <name>pending</name>
 </instanceState>
 <privateDnsName/>
 <dnsName/>
 <reason/>
 <amiLaunchIndex>0</amiLaunchIndex>
 <instanceType>m1.small</instanceType>
 <launchTime>YYYY-MM-DDTHH:MM:SS.SSSZ</launchTime>
 <placement>
 <availabilityZone>us-east-1a</availabilityZone>
 <groupName/>
 <tenancy>default</tenancy>
 </placement>
 <monitoring>
 <state>disabled</state>
 </monitoring>
 <rootDeviceType>ebs</rootDeviceType>
 </item>
 </instancesSet>
</RunInstancesResponse>
```

2. Look for the instance ID in the `instanceId` element in the response and write it down.

You use it to manipulate this instance (including terminating it when you are finished). It takes a few minutes for the instance to launch.

3. Use the [DescribeInstances](#) action to display the launch status of the instance.

Construct the following request.

```
https://ec2.amazonaws.com/
?Action=DescribeInstances
&InstanceId=i-1a2b3c4d
&AUTHPARAMS
```

The following is an example response. In this response, `ami_id` represents the latest version of an AMI.

```
<DescribeInstancesResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
 <reservationId>r-1a2b3c4d</reservationId>
 <ownerId>111122223333</ownerId>
```

```
<groupSet>
 <item>
 <groupId>default</groupId>
 </item>
</groupSet>
<instancesSet>
 <item>
 <instanceId>i-1a2b3c4d</instanceId>
 <imageId>ami-1a2b3c4d</imageId>
 <instanceState>
 <code>16</code>
 <name>running</name>
 </instanceState>
 <privateDnsName>ip-10-251-50-35.ec2.internal</privateDnsName>
 <dnsName>ec2-67-202-51-223.compute-1.amazonaws.com</dnsName>
 <keyName>gsg-keypair</keyName>
 <amiLaunchIndex>0</amiLaunchIndex>
 <instanceType>m1.small</instanceType>
 <launchTime>YYYY-MM-DDTHH:MM:SS.SSSZ</launchTime>
 <placement>
 <availabilityZone>us-east-1a</availabilityZone>
 <groupName/>
 <tenancy>default</tenancy>
 </placement>
 <monitoring>
 <state>disabled</state>
 </monitoring>
 </item>
</instancesSet>
</DescribeInstancesResponse>
```

When the instance state is `running`, as it is in the previous response, the instance has started booting.

## Creating and Launching an AMI from a Snapshot

If you have a snapshot of the root device volume of an instance, you can terminate that instance and then later launch a new instance from the snapshot. You must first register the snapshot, then create and launch the resulting AMI, as explained below.

Launching a new instance from a snapshot is useful if you don't have the original AMI from which to launch new instances. For more information, see [Creating an Amazon EBS Snapshot \(p. 492\)](#).

### Important

At this time, registering a snapshot works only for Linux/UNIX AMIs; although you can register a snapshot to create a Windows AMI, the AMI isn't launchable.

## AWS Management Console

### To launch a Linux/UNIX AMI from a snapshot

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **Elastic Block Store**, click **Snapshots**.
3. Right-click the snapshot from which to create the AMI and click **Create Image from Snapshot**, or click **Create Image** from the toolbar.

4. In the **Create Image from EBS Snapshot** dialog box, complete the fields to create your AMI, then click **Yes, Create**. Your AMI is created and registered.

**Important**

- You must specify the correct architecture for your configuration. Select your architecture (**i386** (32-bit) or **x86\_64** (64-bit)) from the **Architecture** list.
- You must manually specify the AKI to use from the **Kernel ID** list. If you do not specify it here, or if you select the default AKI, you are required to specify the AKI upon every launch. In addition, your instance may fail the health check because the default AKI is incompatible with the instance.

5. In the navigation pane under **IMAGES**, click **AMIs**, click your AMI, then click **Launch**. Follow the additional prompts to define your instance.
6. Click **Launch** to launch your new AMI from your snapshot.

## Command Line tools

### To launch an AMI from a snapshot

1. Use the [ec2-register](#) command and specify a block device mapping that maps the root device name of your choice to the snapshot.

The following example specifies the root device as /dev/sda1, and maps it to the snap-1a2b3c4d snapshot. The resulting root device volume is the same size as the snapshot, and it is automatically be deleted on instance termination. If you specify the block device mapping as /dev/sda1=snap-1a2b3c4d::false, then the volume persists on instance termination.

```
PROMPT> ec2-register -n My_Image_Name -d My_image_description --root-device-name /dev/sda1 -b /dev/sda1=snap-1a2b3c4d --kernel aki-1a2b3c4d
```

The response displays the ID for your new AMI.

```
IMAGE ami-1a2b3c4d
```

The AMI now appears in the list of AMIs that you own. You can view that list in the AWS Management Console, or by using the following command: `ec2-describe-images -o self`.

2. Launch an instance of the AMI.

The resulting instance has a root device volume created from the snapshot.

## API

### To launch an AMI from a snapshot

1. Issue the following Query request to register an image.

The example specifies the root device as /dev/sda1, and maps it to the snap-1a2b3c4d snapshot. The resulting root device volume is the same size as the snapshot, and it is automatically be deleted on instance termination. You can set `DeleteOnTermination` to `false` if you would rather have the volume persist.

```
https://ec2.amazonaws.com/
?Action=RegisterImage
&Name=MyImage
&KernelId=aki-1a2b3c4d
&RamdiskId=ari-1a2b3c4d
&RootDeviceName=/dev/sda1
&BlockDeviceMapping.1.DeviceName=/dev/sda1
&BlockDeviceMapping.1.Ebs.SnapshotId=snap-1a2b3c4d
&AUTHPARAMS
```

For information about the auth parameters, see [Common Query Parameters](#) in the *Amazon Elastic Compute Cloud API Reference*.

The following is an example response.

```
<RegisterImageResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
 <requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId>
 <imageId>ami-1a2b3c4d</imageId>
</RegisterImageResponse>
```

The AMI now appears in the list of AMIs that you own. You can view that list by using `DescribeImages` with `Owner=self`.

2. Launch an instance of the AMI.

The resulting instance has a root device volume created from the snapshot.

# Connecting to Amazon EC2 Instances

This section describes how to connect to instances that you launched and how to transfer files between your local computer and your Amazon EC2 instance. To connect to an instance, you will need a running instance and the private key file that corresponds to the key pair that was selected when the instance was launched. For information about launching instances, see [Launching EC2 Instances \(p. 274\)](#). For information about key pairs, see [Getting a Key Pair \(p. 274\)](#).

The following topics demonstrate how to connect to your instance:

## Topics

- [Authorizing Network Access to Your Instances \(p. 293\)](#)
- [Connecting to Your Linux/UNIX Instances Using SSH \(p. 295\)](#)
- [Connecting to Linux/UNIX Instances from Windows Using PuTTY \(p. 301\)](#)
- [Connecting to Windows Instances \(p. 306\)](#)

## Authorizing Network Access to Your Instances

Your AWS account comes with a default security group named `default`. This security group does not permit access on any ports, so if your instance is associated with that security group, you will not be able to connect to it. To access your instance with SSH or RDP, your instance must allow incoming traffic on port 22 or 3389, respectively. To open a port for incoming traffic, add a security group rule to a security group that's associated with your instance. You can use the AWS Management Console or the command line tools. If you use the command line tools, use them on your local system, not on the instance itself.

The following instructions authorize incoming SSH or RDP traffic for your instance, but only from your local system's public IP address. If your IP address is dynamic, you must authorize access each time it changes. To allow additional IP address ranges, add a new security group rule for each range.

### Important

If you used the Launch Wizard to launch your instance, and it automatically created the `quicklaunch-x` or `quick-start-x` security group for you, be aware that these security groups allow all incoming SSH or RDP traffic by default. For security reasons, it is not recommended that you allow access from all IP addresses (0.0.0.0/0) to your instance, so you should update the rules in these security groups as soon as possible. For more information about security groups, see [Amazon EC2 Security Groups \(p. 415\)](#).

### Before You Get Started

Decide who requires access to your instance, for example, a single host or a specific network that you trust. In this case, we will use your local system's public IP address. You can get the public IP address of your local computer using a service. To locate a service that provides your IP address, use the search phrase "what is my IP address". If you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

## AWS Management Console

### To add a rule to a security group for SSH access for Linux instances

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Instances**. Select your instance and look at the **Description** tab; the **Security Groups** field lists the security groups that are associated with the instance. Click the **view rules** link to display a list of the rules that are in effect for the instance.
3. In the navigation pane, click **Security Groups**. Select one of the EC2 security groups associated with your instance.

4. On the **Inbound** tab, select **SSH** from the **Create a new rule** drop-down list.



5. In the **Source** field, specify your local system's public IP address in CIDR notation. For example, if your IP address is 203.0.113.0, enter 203.0.113.0/32.
6. Click **Add Rule**.

An asterisk appears on the **Inbound** tab, indicating that the rule hasn't been applied.

7. When you're finished adding rules, click **Apply Rule Changes**.

The new rules are applied to all instances that are associated with the security group.

#### To add a rule to a security group for RDP access for Windows instances

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Instances**. Select your instance and look at the **Description** tab; the **Security Groups** field lists the security groups that are associated with the instance. Click the **view rules** link to display a list of the rules that are in effect for the instance.
3. In the navigation pane, click **Security Groups**. Select one of the EC2 security groups associated with your instance.
4. On the **Inbound** tab, select **RDP** from the **Create a new rule** drop-down list.



5. In the **Source** field, specify your local system's public IP address in CIDR notation. For example, if your IP address is 203.0.113.0, enter 203.0.113.0/32.
6. Click **Add Rule**.

An asterisk appears on the **Inbound** tab, indicating that the rule hasn't been applied.

7. When you're finished adding rules, click **Apply Rule Changes**.  
The new rules are applied to all instances that are associated with the security group.

## Command Line Interface

Use the `ec2-authorize` command. For information about the command, see [ec2-authorize](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

### To add a rule to a security group for SSH access

- The following command adds a rule to the default security group that allows incoming traffic on port 22 (SSH port) from your IP address.

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

### To add a rule to a security group for RDP access

- The following command opens port 3389 (RDP port) to your IP address.

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

## API

Use the [AuthorizeSecurityGroupIngress](#) action.

## Connecting to Your Linux/UNIX Instances Using SSH

To connect to an instance, you will need a running instance and the private key file that corresponds to the key pair that was selected when the instance was launched. For information about launching instances, see [Launching EC2 Instances \(p. 274\)](#). For information about key pairs, see [Getting a Key Pair \(p. 274\)](#).

### Topics

- [Connecting to Your Instance from Your Web Browser Using a Java-Based SSH Client \(p. 295\)](#)
- [Connecting to Linux/UNIX Instances from Linux/UNIX using SSH \(p. 297\)](#)
- [Transferring Files to Linux/UNIX Instances from Linux/UNIX with SCP \(p. 299\)](#)

## Connecting to Your Instance from Your Web Browser Using a Java-Based SSH Client

### Prerequisites

- Install Java (see [http://java.com/en/download/help/index\\_installing.xml](http://java.com/en/download/help/index_installing.xml))

On a Windows or Mac client, you must run your Web browser with administrator credentials. For Linux, additional steps may be required if you are not logged in as root.

- Enable Java in your web browser (see [http://java.com/en/download/help/enable\\_browser.xml](http://java.com/en/download/help/enable_browser.xml))

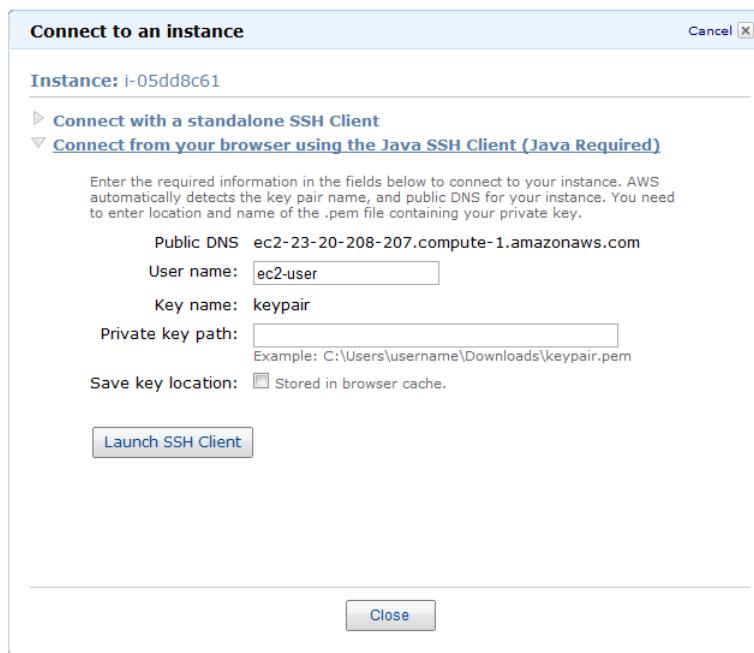
### To connect to your instance using a web browser with an SSH client

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Instances**.
3. Right-click your instance, and select **Connect**.
4. Click **Connect from your browser using the Java SSH client (Java Required)**. AWS automatically detects the public DNS address of your instance and the key pair name you launched the instance with.
5. In **User name**, enter the user name to log in to your instance.

**Note**

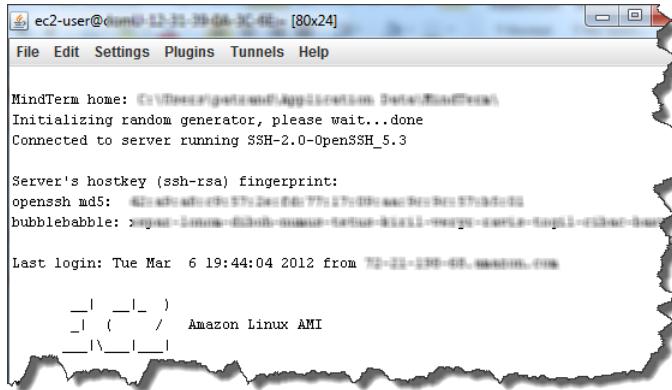
For an Amazon Linux instance, the default user name is `ec2-user`. For Ubuntu, the default user name is `ubuntu`. Some AMIs allow you to log in as root.

6. The **Key name** field is automatically populated for you.
7. In **Private key path**, enter the fully qualified path to your .pem private key file.
8. Click **Save key location**, click **Stored in browser cache** to store the key location in your browser cache so the key location is detected in subsequent browser sessions, until you clear your browser's cache.
9. Click **Launch SSH Client**.



10. If necessary, click **Yes** to trust the certificate.
11. Click **Run** to run the MindTerm client.
12. If you accept the license agreement, click **Accept**.
13. If this is your first time running MindTerm, a series of dialog boxes will ask you to confirm setup for your home directory and other settings.

14. Confirm settings for MindTerm setup. A screen opens and you are connected to your instance.



## Connecting to Linux/UNIX Instances from Linux/UNIX using SSH

### Prerequisites

- **Install an SSH client**

Most Linux and UNIX computers include an SSH client by default. You can check for an SSH client by typing **ssh** at the command line. If your computer doesn't recognize the command, the OpenSSH project provides a free implementation of the full suite of SSH tools. For more information, see <http://www.openssh.org>.

- **Amazon EC2 CLITools**—Install the tools to enable use of the `ec2-get-console-output` command.

For more information, see [Setting Up the Amazon EC2 Command Line Interface Tools on Linux/UNIX \(p. 560\)](#).

- **Private key**—Get the path to your private key

You'll need the fully qualified path of the private key file associated with your instance. For more information on key pairs, see [Getting a Key Pair \(p. 274\)](#).

- **Enable SSH traffic**—Open the SSH port on the instance.

Before you try to connect, ensure that your instance accepts incoming SSH traffic (usually on port 22). For more information, see [Authorizing Network Access to Your Instances \(p. 293\)](#).

### To connect to your instance using SSH

1. If you've launched a public AMI, run the `ec2-get-console-output` command on your local system (not on the instance), and locate the **SSH HOST KEY FINGERPRINTS** section. For more information, see [ec2-get-console-output](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

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

...
ec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----
ec2: 2048 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
/etc/ssh/ssh_host_key.pub
ec2: 2048 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
/etc/ssh/ssh_host_rsa_key.pub
```

```
ec2: 1024 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:
/etc/ssh/ssh_host_dsa_key.pub
ec2: -----END SSH HOST KEY FINGERPRINTS-----
...
```

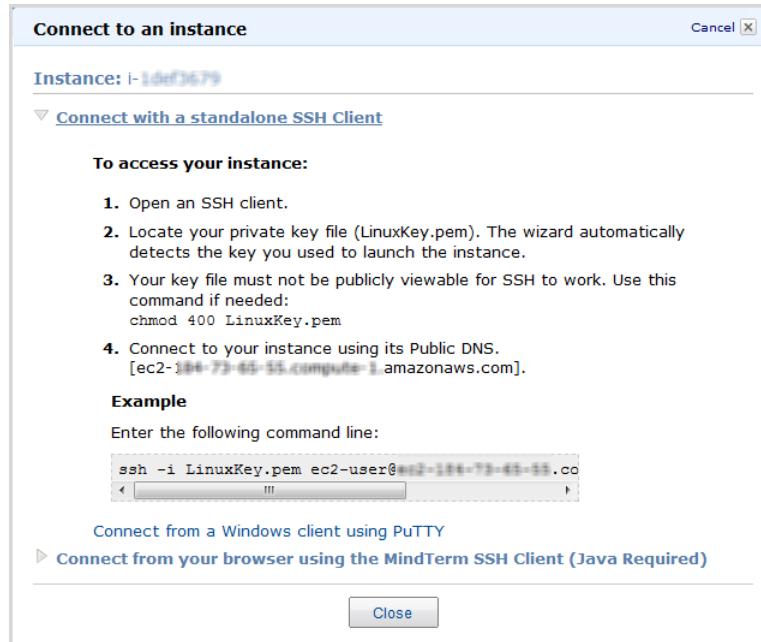
Note the fingerprints so that you can compare them to the fingerprints of the instance.

2. In a command line shell, change directories to the location of the private key file that you created when you launched the instance.
3. Use the `chmod` command to make sure your private key file isn't publicly viewable. For example, if your private key file were `My_Keypair.pem`, you would enter:

```
chmod 400 My_Keypair.pem
```

4. In the navigation pane, click **Instances**.
5. Right-click your instance, and then click **Connect**.
6. Click **Connect using a standalone SSH client**. AWS automatically detects the public DNS address of your instance and the key pair name you launched the instance with.
7. Copy the example command provided in the Amazon EC2 console if you launched an Amazon Linux instance. If you used a different Amazon Machine Image (AMI) for your Linux/UNIX instance, you need to log in as the default user for the AMI. For an Ubuntu instance, the default user name is `ubuntu`. Some AMIs allow you to log in as `root` so you will need to change the user name from `ec2-user` to the appropriate user name.

```
ssh -i <your key a name>.pem ec2-user@ec2-184-72-204-112.compute-1.amazonaws.com
```



You'll see a response like the following.

```
The authenticity of host 'ec2-184-72-204-112.compute-1.amazonaws.com
(10.254.142.33)'

can't be established.

RSA key fingerprint is 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00.

Are you sure you want to continue connecting (yes/no)? yes
```

**Important**

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

8. Enter `yes`.

You'll see a response like the following.

```
Warning: Permanently added 'ec2-184-72-204-112.compute-1.amazonaws.com'
(RSA)

to the list of known hosts.
```

## Transferring Files to Linux/UNIX Instances from Linux/UNIX with SCP

One way to transfer files between your local computer and a Linux/UNIX instance is to use Secure Copy (SCP). This section describes how to transfer files with SCP. The procedure is very similar to the procedure for connecting to an instance with SSH.

### Prerequisites

- **Enable SSH traffic**—Open the instance's SSH port  
Before you try to connect, ensure that your Amazon EC2 instance accepts incoming SSH traffic (usually on port 22). For more information, see [Authorizing Network Access to Your Instances \(p. 293\)](#).
- **SCP client**—Install an SCP client  
Most Linux and UNIX computers include an SCP client by default. If yours doesn't, the OpenSSH project provides a free implementation of the full suite of SSH tools, including an SCP client. For more information, go to <http://www.openssh.org>.
- **Instance ID**—Get the ID of your Amazon EC2 instance  
Retrieve the Instance ID of the Amazon EC2 instance you want to access. The Instance IDs for all of your instances are available in the Amazon EC2 console or with the `ec2-describe-instances` command.
- **Instance's public DNS**—Get the public DNS of your Amazon EC2 instance  
Retrieve the public DNS of the Amazon EC2 instance you want to access. You can find the public DNS for your instance using the Amazon EC2 console or with the `ec2-describe-instances` command.
- **Private key**—Get the path to your private key  
You'll need the fully qualified path of the private key file associated with your instance. For more information on key pairs, see [Getting a Key Pair \(p. 274\)](#).

The following procedure steps you through using SCP to transfer a file. If you've already connected to the instance with SSH and have verified its fingerprints, you can start with the step that contains the SCP command (step 4).

## To use SCP to transfer a file

- If you've launched a public AMI, run the `ec2-get-console-output` command on your local system (not on the instance) and locate the **SSH HOST KEY FINGERPRINTS** section. For more information, see [ec2-get-console-output](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

```
PROMPT> ec2-get-console-output instance_id
...
ec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----
ec2: 2048 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:
/etc/ssh/ssh_host_key.pub
ec2: 2048 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:
/etc/ssh/ssh_host_rsa_key.pub
ec2: 1024 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:
/etc/ssh/ssh_host_dsa_key.pub
ec2: -----END SSH HOST KEY FINGERPRINTS-----
...
```

Note the fingerprints so that you can compare them to the fingerprints of the instance.

2. In a command line shell, change directories to the location of the private key file that you created when you launched the instance.
  3. Use the `chmod` command to make sure your private key file isn't publicly viewable. For example, if your file were `My Keypair.pem`, you would enter:

```
chmod 400 My_Keypair.pem
```

- Transfer a file to your instance using the instance's public DNS name (available through the AWS Management Console or the `ec2-describe-instances` command). For example, if the key file is `My Keypair.pem`, the file to transfer is `samplefile.txt`, and the instance's DNS name is `ec2-184-72-204-112.compute-1.amazonaws.com`, use the following command to copy the file to the `ec2-user` home directory.

```
scp -i My_Keypair.pem samplefile.txt ec2-user@ec2-184-72-204-112.compute-1.amazonaws.com:~
```

## Note

Some AMIs let you log in as root, but some require that you log in with the username ec2-user. For log in information for your chosen AMI, contact your AMI provider directly or see [Amazon Machine Images \(AMIs\)](#) page, then locate and click your AMI on the list.

You'll see a response like the following.

```
The authenticity of host 'ec2-184-72-204-112.compute-1.amazonaws.com
(10.254.142.33)'
can't be established.
RSA key fingerprint is 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00.
Are you sure you want to continue connecting (yes/no)? yes
```

## **Important**

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

5. Enter yes.

You'll see a response like the following.

```
Warning: Permanently added 'ec2-184-72-204-112.compute-1.amazonaws.com' (RSA)
to the list of known hosts.
Sending file modes: C0644 20 samplefile.txt
Sink: C0644 20 samplefile.txt
samplefile.txt 100% 20 0.0KB/s 00:00
```

To transfer files in the other direction, i.e., from your Amazon EC2 instance to your local computer, simply reverse the order of the host parameters. For example, to transfer the samplefile.txt file from your Amazon EC2 instance back to the home directory on your local computer as samplefile2.txt, use the following command on your local computer.

```
scp -i My_Keypair.pem ec2-user@ec2-184-72-204-112.compute-1.amazonaws.com:~/samplefile.txt ~/samplefile2.txt
```

## Connecting to Linux/UNIX Instances from Windows Using PuTTY

To connect to an instance, you will need a running instance and the private key file that corresponds to the key pair that was selected when the instance was launched. For information about launching instances, see [Launching EC2 Instances \(p. 274\)](#). For information about key pairs, see [Getting a Key Pair \(p. 274\)](#).

### Topics

- [Getting PuTTY \(p. 302\)](#)
- [Converting Your Private Key \(p. 302\)](#)
- [Connecting Using PuTTY SSH \(p. 303\)](#)
- [Transferring Files with PSCP \(p. 306\)](#)

To connect to your Linux/UNIX instance from a Windows computer, use an SSH client. The following instructions explain how to use PuTTY, a free SSH client for Windows computers.

### Prerequisites

- **Enable SSH traffic**—Open the instance's SSH port  
Before you try to connect, ensure that your Amazon EC2 instance's security group accepts incoming SSH traffic on port 22. For more information, see [Authorizing Network Access to Your Instances \(p. 293\)](#).
- **Instance ID**—Get the ID of your Amazon EC2 instance  
Retrieve the Instance ID of the Amazon EC2 instance you want to access. The Instance IDs for all of your instances are available in the Amazon EC2 console or with the `ec2-describe-instances` command.
- **Instance's public DNS**—Get the public DNS of your Amazon EC2 instance  
Retrieve the public DNS of the Amazon EC2 instance you want to access. You can find the public DNS for your instance using the Amazon EC2 console or with the `ec2-describe-instances` command.
- **Private key**—Get the path to your private key  
You'll need the fully qualified path of the private key file associated with your instance. For more information on key pairs, see [Getting a Key Pair \(p. 274\)](#).

## Getting PuTTY

### To download and install PuTTY

- Go to <http://www.chiark.greenend.org.uk/~sgtatham/putty/> and follow the instructions there.

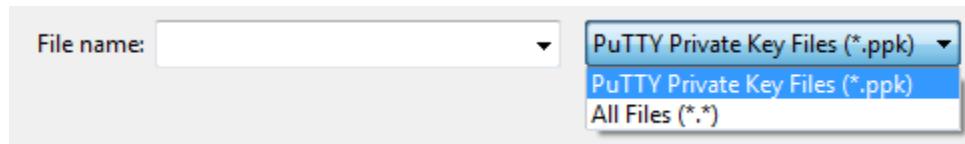
Other tools in the PuTTY suite are PuTTYgen, a key generation program, and pscp, a secure copy command line tool. The different PuTTY tools are separate applications. You can install them separately or install the entire suite with a simple Windows installer. The following instructions assume you've installed the entire suite and can access all the components from the Windows Start menu.

## Converting Your Private Key

PuTTY does not natively support the private key format generated by Amazon EC2. Fortunately, PuTTY has a tool called PuTTYgen, which can convert keys to the required PuTTY format.

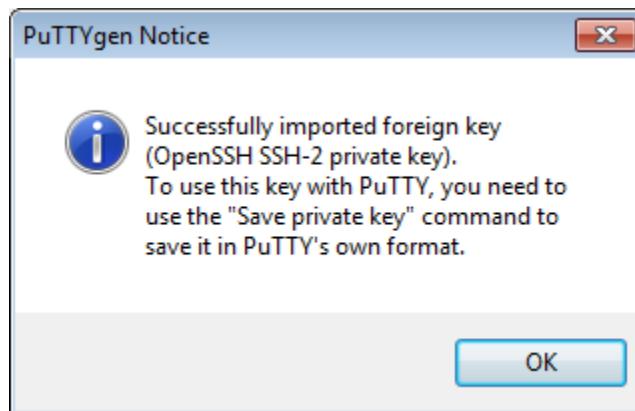
### To convert your private key

1. Start PuTTYgen (for example, from the **Start** menu, click **All Programs > PuTTY > PuTTYgen**).
2. Click **Load** and browse to the location of the private key file that you want to convert (e.g., `gsg-keypair.pem`). By default, PuTTYgen displays only files with extension `.ppk`; you'll need to change that to display files of all types in order to see your `.pem` key file. The private key file must end with a newline character or PuTTYgen cannot load it correctly.



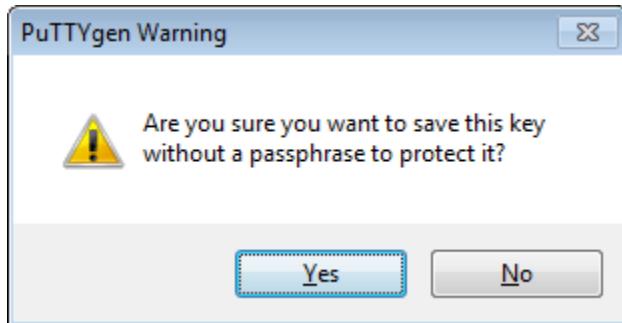
3. Select your `.pem` key file and click **Open**.

PuTTYgen displays the following message.



When you click **OK**, PuTTYgen displays a dialog box with information about the key you loaded, such as the public key and the fingerprint. The keys that Amazon EC2 generates are 1024-bit SSH-2 RSA keys.

4. Click **Save private key** to save the key in PuTTY's format.  
PuTTYgen asks if you want to save the key without a passphrase.



5. Click **Yes**.

**Note**

A passphrase on a private key is an extra layer of protection, so even if your private key is discovered, it will not be usable without the passphrase. The downside to using a passphrase is that it makes automation harder because human intervention is needed to log on to an instance, or copy files to an instance. For this exercise, we're not using a passphrase.

6. Use the same name for the key that you used for the key pair (for example, `gsg-keypair`). PuTTY automatically adds the `.ppk` file extension.

Your private key is now in the correct format for use with PuTTY. You can now connect to your instance using PuTTY's SSH client.

## Connecting Using PuTTY SSH

You'll connect by starting a PuTTY SSH session.

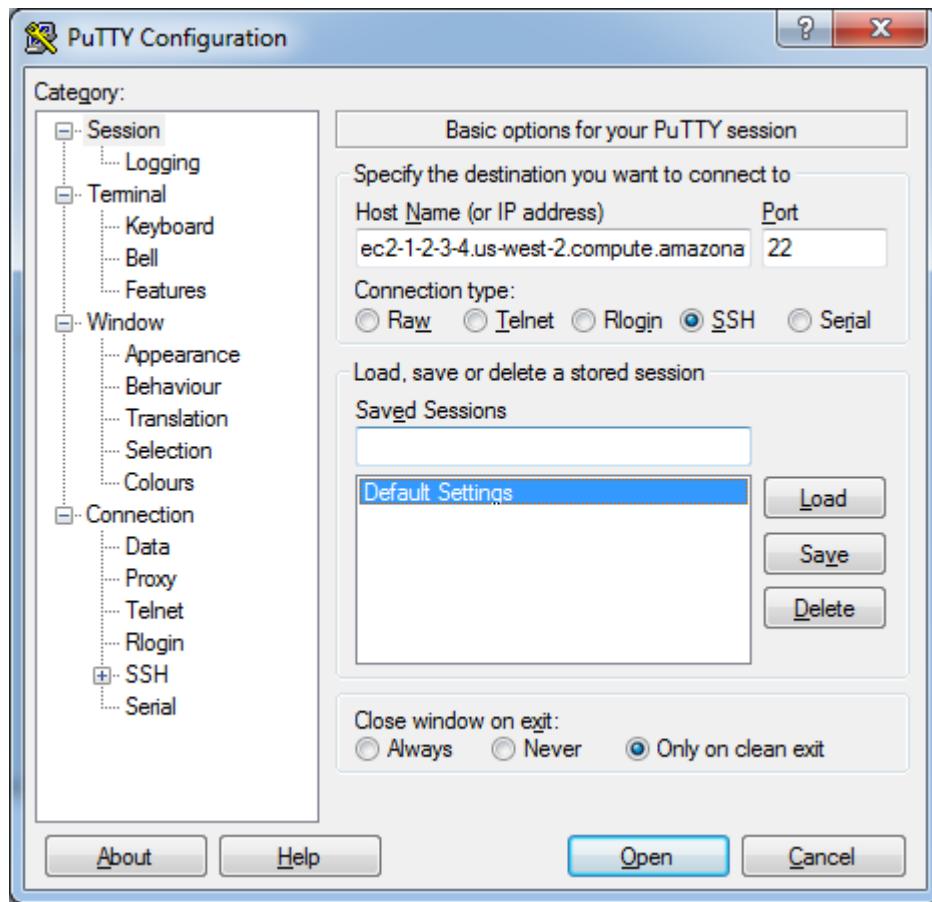
### To use SSH to connect

1. If you've launched a public AMI that isn't one that you created yourself, run the `ec2-get-console-output` command on your local system (not on the instance), and locate the **SSH HOST KEY FINGERPRINTS** section. For more information, see [ec2-get-console-output](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

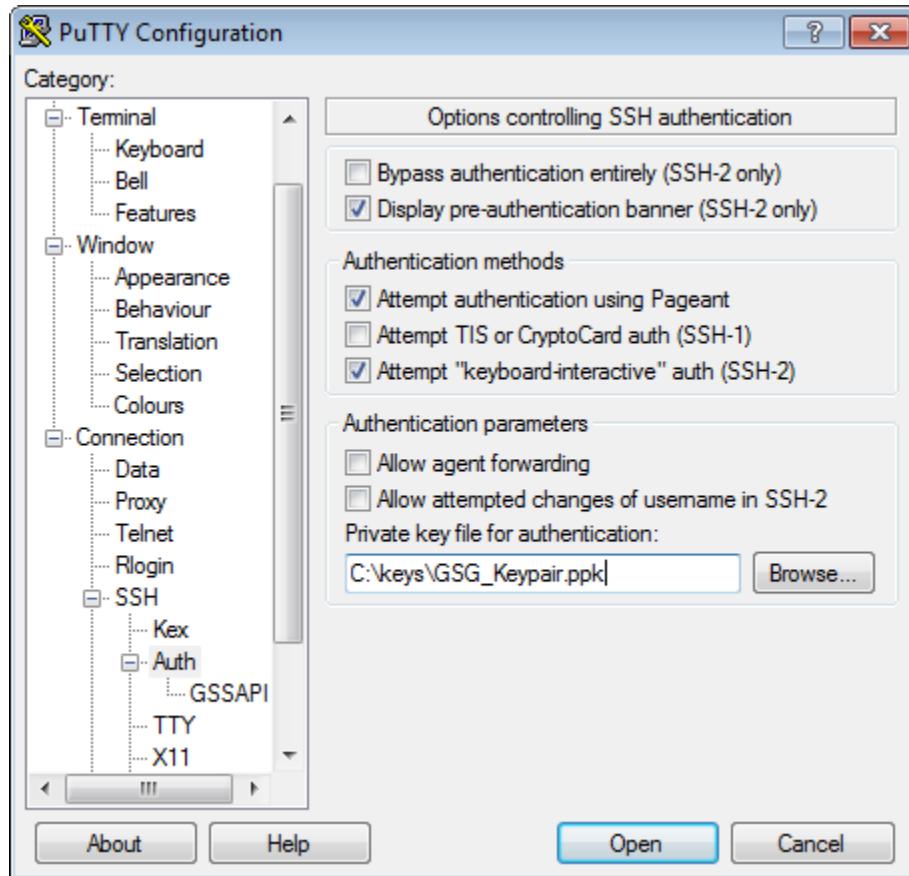
```
C:\> ec2-get-console-output instance_id
...
ec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----
ec2: 2048 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
/etc/ssh/ssh_host_key.pub
ec2: 2048 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
/etc/ssh/ssh_host_rsa_key.pub
ec2: 1024 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
/etc/ssh/ssh_host_dsa_key.pub
ec2: -----END SSH HOST KEY FINGERPRINTS-----
...
```

Note the fingerprints so that you can compare them to the fingerprints of the instance.

2. Start PuTTY (from the **Start** menu, click **All Programs > PuTTY > PuTTY**).
3. In the **Host Name** field, enter the public DNS name of your instance (available through the Amazon EC2 console or the [ec2-describe-instances](#) command).



4. Under **Connection type**, select **SSH** and ensure that the **Port** is set to 22.
5. In the **Category** tree, under **Connection**, expand the **SSH** node, and then select **Auth**.
6. Click **Browse...** and select the PuTTY private key file you generated in the preceding section.



7. If you will be opening this same session later, you may want to save it for future use. To save the session information, select **Session** in the **Category** tree, enter a name for the session in **Saved Sessions**, and click **Save**. This step is optional.
8. Click **Open**. If this is the first time you have connected to this instance, a security alert dialog box will be displayed asking if you trust the host you are connecting to,

**Important**

If you've launched a public AMI, verify that the fingerprint in the security alert matches the fingerprint from the output of the `ec2-get-console-output` command. If it doesn't, someone might be attempting a "man-in-the-middle" attack.

9. If the security alert dialog box is displayed, click **Yes**.
10. In the SSH session window, log in with the appropriate username. In most cases, the username will be `ec2-user`. No password is required because the authentication is provided by the selected public/private key pair.

**Note**

Some AMIs let you log in as root, but some require you to log in with the username `ec2-user`. For log in information for your chosen AMI, contact your AMI provider directly or see the [Amazon Machine Images \(AMIs\)](#) page, then locate and click your AMI on the list.

**Note**

If you specified a passphrase when you converted your private key to PuTTY's format, you must provide that passphrase when you log in to the instance.

If you receive a "Connection Refused" error when connecting to the instance, ensure that your Amazon EC2 instance's security group accepts incoming SSH traffic on port 22. For more information, see [Authorizing Network Access to Your Instances \(p. 293\)](#).

## Transferring Files with PSCP

The PuTTY Secure Copy Client (PSCP) is a command-line tool that lets you transfer files between your Windows computer and your Linux/UNIX instance.

To use PSCP, you'll need the private key you generated in [Converting Your Private Key \(p. 302\)](#). You'll also need the public DNS address of your Linux/UNIX instance.

The following example transfers the file `sample_file.txt` from a Windows computer to the `/usr/local` directory on a Linux/UNIX instance:

```
C:\> pscp -i C:\Keys\gsg-keypair.ppk C:\sample_file.txt username@public-DNS:/usr/local/sample_file.txt
```

If you prefer a graphical user interface (GUI), you can use an open source GUI tool named WinSCP. For more information, see the [WinSCP website](#).

## Connecting to Windows Instances

To connect to an instance, you will need a running instance and the private key file that corresponds to the key pair that was selected when the instance was launched. For information about launching instances, see [Launching EC2 Instances \(p. 274\)](#). For information about key pairs, see [Getting a Key Pair \(p. 274\)](#).

This section describes how to connect to instances running Windows from local computers running Windows, Linux/UNIX, or Mac OS.

### Prerequisites

- **Enable RDP traffic**—Open the instance's RDP port  
Before you try to connect, ensure that your Amazon EC2 instance accepts incoming RDP traffic (usually on port 3389). For more information, see [Authorizing Network Access to Your Instances \(p. 293\)](#).
- **RDP client**—Install an RDP client  
Windows computers include an RDP client by default. You can check for an RDP client by typing `mstsc` at a Command Prompt window. If your computer doesn't recognize this command, see the [Microsoft Windows home page](#) and search for the download for Remote Desktop Connection. For Mac OS X, you can use [Microsoft's Remote Desktop Client](#). For Linux/UNIX, you can use `rdesktop`.
- **Instance ID**—Get the ID of your Amazon EC2 instance  
Retrieve the Instance ID of the Amazon EC2 instance you want to access. The Instance IDs for all of your instances are available in the Amazon EC2 console or with the `ec2-describe-instances` command.
- **Private key**—Get the path to your private key  
You'll need the fully qualified path of the private key file associated with your instance. For more information on key pairs, see [Getting a Key Pair \(p. 274\)](#).

## Connecting to Amazon EC2 Windows Instances

To connect to a Windows instance, you must retrieve the initial administrator password, and then specify this password with Remote Desktop. You'll need the private key file that you created when you launched the instance (for example, `GSG_Keypair.pem`).

## To connect to your Windows instance

1. Before you try to connect, ensure that your Amazon EC2 instance accepts incoming RDP traffic (usually on port 3389). For more information, see [Authorizing Network Access to Your Instances](#).
2. Windows computers include an RDP client by default. You can check for an RDP client by typing **mstsc** at the Command Prompt window. If your computer doesn't recognize this command, go to the [Microsoft Windows home page](#) and search for the download for Remote Desktop Connection. For Mac OS X, you can use [Microsoft's Remote Desktop Client](#). For Linux/UNIX, you can use [rdesktop](#).
3. In the Amazon EC2 console, right-click the instance that you created and click **Connect**.
4. In the **Console Connect** dialog box, click **Retrieve Password** (it will take a few minutes after the instance is launched before the password is available).
5. Click **Browse** and navigate to the private key file you created when you launched the instance. Select the file and click **OK** to copy the entire contents of the file into the **Private Key contents** box.
6. Click **Decrypt Password**. The console displays the default administrator password for the instance in the **Console Connect** dialog box, replacing the link to **Retrieve Password** shown previously with the actual password.
7. Record the default administrator password, or copy it to the clipboard. You need this password to connect to the instance.
8. Click **Download shortcut file**. Your browser prompts you to either open or save the .rdp file. Either option is fine. When you have finished, you can click **Close** to dismiss the **Console Connect** dialog box.
9. If you opened the .rdp file, you'll see the **Remote Desktop Connection** dialog box. If you saved the .rdp file, navigate to your downloads directory, and double-click the .rdp file to display the dialog box. You may get a warning that the publisher of the remote connection is unknown. Click **Connect** to connect to your instance. You may get a warning that the security certificate could not be authenticated. Click **Yes** to continue.
10. Log in to the instance as prompted, using `Administrator` as the user name and the default administrator password that you recorded or copied in step 7.

We recommend that you do the following:

- Change the Administrator password from the default value. You change the password while logged on to the instance itself, just as you would on any other Windows Server.
- Create another user account with administrator privileges on the instance. Another account with administrator privileges is a safeguard if you forget the Administrator password or have a problem with the Administrator account.

### Note

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

## Transfer Files to Windows Server Instances from Windows

You can work with your instance the same way you would work with any Windows server. For example, you can transfer files between an Amazon EC2 Windows instance and your local Windows computer using the local file sharing feature of Windows Remote Desktop. If you enable this option in your Windows Remote Desktop Connection software, you can access your local files from your Amazon EC2 Windows instances. You can access local files on hard disk drives, DVD drives, portable media drives, and mapped network drives. For information about this feature, go to the [Microsoft Support website](#) or go to [The most useful feature of Remote Desktop I never knew about](#) on the MSDN Blogs website.

# Importing and Exporting Instances

You can import a virtual machine (VM) from a Citrix Xen, Microsoft Hyper-V, or VMware vSphere virtualization platform and then launch it in Amazon EC2. Later, you can export that Amazon EC2 instance back to Citrix Xen, Microsoft Hyper-V, or VMware vSphere.

## Topics

- [Importing EC2 Instances \(p. 308\)](#)
- [Exporting EC2 Instances \(p. 338\)](#)

## Importing EC2 Instances

### Topics

- [Components in Your VM Environment \(p. 309\)](#)
- [Before You Get Started \(p. 309\)](#)
- [Using the Amazon EC2 VM Import Connector to Import Your Virtual Machine to Amazon EC2 \(p. 310\)](#)
- [Using the Command Line Tools to Import Your Virtual Machine to Amazon EC2 \(p. 324\)](#)
- [Troubleshooting Instance Importation \(p. 336\)](#)

There are two ways you can launch an instance in the Amazon Elastic Compute Cloud (Amazon EC2). You can launch an instance from an AMI that you created or selected from a catalog. Or, you can launch an instance from a virtual machine (VM) that you imported from a Citrix Xen, Microsoft Hyper-V, VMware Workstation, or VMware vSphere virtualization environment. This section covers using VMs from Citrix, Microsoft, or VMware to launch instances.

To use your virtual machine as an instance in Amazon EC2, you must first export it from the virtualization environment using its tools. Then you import it to Amazon EC2 using the Amazon EC2 command line or API tools.

If you are importing a VMware vSphere VM, you can also use the Amazon EC2 VM Import Connector for VMware (Connector), a plug-in that integrates with VMware vSphere Client, to perform the task.

Whether you use the command line tools, the API, or the Connector, you will follow the same general process for importing VMs or volumes to Amazon EC2. You need to complete these tasks, which are all discussed in this section:

1. Prepare the virtual machine for import to Amazon EC2. For more information, go to [Before You Get Started \(p. 309\)](#)
2. Export the virtual machine from the virtualization environment.
  - [Exporting from Citrix \(p. 325\)](#)
  - [Exporting from Microsoft Hyper-V \(p. 328\)](#)
  - [Exporting from VMware \(p. 329\)](#)
3. Import the virtual machine to Amazon EC2.  
For information about using the command line tools to import your VM, see [Using the Command Line Tools to Import Your Virtual Machine to Amazon EC2 \(p. 324\)](#). For information about using the Connector, see [Using the Amazon EC2 VM Import Connector to Import Your Virtual Machine to Amazon EC2 \(p. 310\)](#).
4. Upload the instance to Amazon EC2.
5. Launch the instance in Amazon EC2.

## Components in Your VM Environment

The table in this section describes the typical components in your VM environment.

Component	Description	Product Name
Virtualization product	Virtualization service for managing virtual computing infrastructure	Citrix Xen Microsoft Hyper-V VMware Workstation VMware vSphere (vSphere)
Client	The software you need on your computer to access and manage your virtualization environment	Citrix Xen Center Microsoft Hyper-V Manager VMware Workstation VMware vSphere Client
Server	The management platform for the virtualization environment	Citrix XenServer Microsoft Hyper-V VMware vCenter Server
Amazon EC2 VM Import Connector for VMware (Connector)	The virtual appliance, a plug-in to the management platform of the virtualization Server, that enables the import of virtual machines into Amazon EC2 using the Client interface	Citrix Xen (not applicable) Microsoft Hyper-V (not applicable) VMware vCenter—Amazon EC2 VM Import Connector for VMware vCenter (Connector)

## Before You Get Started

This section discusses the things you need to know and what you must have before you begin the process of importing your virtual machine.

**Operating Systems**—The following operating systems can be imported to Amazon EC2:

- Microsoft Windows Server 2003 (Standard, Datacenter, Enterprise).
- Microsoft Windows Server 2003 R2 (Standard, Datacenter, Enterprise).
- Microsoft Windows Server 2008 (Standard, Datacenter, Enterprise).
- Microsoft Windows Server 2008 R2 (Standard, Datacenter, Enterprise).

**Image Formats**—We support import of the following image formats for importing both volumes and instances to Amazon Web Services:

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

**Known Limitations**—The importing of instances and volumes is subject to the following limitations:

- You can have up to five conversions and tasks in progress at the same time per Region.
- Typically, you import a compressed version of a disk image; the expanded image cannot exceed 1TB.

- Tasks must complete within 7 days of the start date.
- Importing virtual machines with more than one virtual disk is not supported. We suggest that you import the VM with only the boot volume, and import any additional disks using `ImportVolume` (`ec2-import-volume`) in the command line. After the `ImportInstance` task is complete, use `AttachVolume` (`ec2-attach-volume`) to associate the additional volumes with your instance.
- Multiple network interfaces are not currently supported. When converted and imported, your instance will have a single virtual NIC using DHCP for address assignment.
- For vCenter 4.0 and vSphere 4.0 users, remove any attached CD-ROM images or ISOs from the virtual machine.

## Preparing Your Virtual Machine

Use the following guidelines to configure your virtual machine before exporting it from the virtualization environment.

### Important

If you are importing a virtual machine from Citrix Xen, you must uninstall the Citrix Tools for Virtual Machines from the VM. If you don't uninstall the tools, your import will fail. For more information, see [Exporting from Citrix \(p. 325\)](#).

- Enable Remote Desktop (RDP) for remote access.
- Make sure your host firewall (Windows firewall), if configured, allows access to RDP. Otherwise, you will not be able to access your instance after the conversion is complete.
- Make sure all user accounts use secure passwords. This includes the administrator account.

### Note

All accounts must have passwords. Otherwise, the import might fail.

- Disable any antivirus or intrusion detection software on your virtual machine. These services can be re-enabled after the import process is complete.
- Disconnect any CD-ROM drives (virtual or physical).
- Do not Sysprep your virtual machine images. We recommend that you import the image and then use the Amazon EC2 Config service to Sysprep it.
- Set your network to DHCP. If you are using a private IP address, be sure to use a non-reserved private IP address in your VPC subnet. Amazon Virtual Private Cloud (Amazon VPC) reserves the first four private IP address in a VPC subnet.
- Shut down your virtual machine before exporting it.
- Make sure you've installed .NET Framework 3.5, as required by [Amazon Windows EC2Config Service](#).

## Using the Amazon EC2 VM Import Connector to Import Your Virtual Machine to Amazon EC2

### Topics

- [Before You Install the Connector \(p. 311\)](#)
- [Installing the Connector for VMware vCenter \(p. 312\)](#)
- [Configuring the Connector for VMware vCenter \(p. 316\)](#)
- [Using the Connector for VMware vCenter \(p. 321\)](#)
- [Uninstalling the Connector for VMware vCenter \(p. 323\)](#)

You can use the Amazon EC2 VM Import Connector virtual appliance (vApp), a plug-in for VMware vCenter, to import virtual machines from your VMware vSphere infrastructure to Amazon EC2. The

Connector is a virtual appliance that works with VMware vCenter Server only. It provides an easy-to-use interface, enhancing your existing management tools to work with the Amazon EC2 VM Import Connector.

**Note**

You cannot use the Connector to import Citrix Xen or Microsoft Hyper-V virtual machines to Amazon EC2. Instead, use the command line tools to import your Citrix and Hyper-V virtual machines to Amazon EC2. You can also choose to use the command line tools to import your VMware VMs. For more information, see [Using the Command Line Tools to Import Your Virtual Machine to Amazon EC2 \(p. 324\)](#).

By comparing the following procedures, you can see that using the Connector simplifies the process of importing your VMware VMs.

**To import VMware VMs using the Connector**

1. Import the VM to Amazon EC2
2. Launch the instance

**To import VMware VMs using Amazon EC2**

1. Export the VM from the virtualization environment
2. Import your VM to Amazon EC2
3. Upload the instance to Amazon EC2
4. Launch the instance in Amazon EC2

**Before You Install the Connector**

To use the VM Import Connector, you first need to install the Connector virtual appliance. Before you install, read through the general prerequisites listed in the [Before You Get Started \(p. 309\)](#) section and make sure that your virtualization environment meets the following requirements:

**VMware infrastructure**

- vSphere 4.0, 4.1, or 5.0
- vCenter 4.0, 4.1, or 5.0

**Amazon EC2 VM Import Connector vApp**

- 256MB RAM
- Minimum 250GB of disk space

**Note**

Although the Connector virtual appliance is small, it temporarily stores the VM images that you import to Amazon EC2. The data store must be large enough to accommodate these images, plus the Connector. We recommend a data store size of 250GB or larger.

To estimate the disk space you need, multiply the maximum number of parallel import tasks you want to run with the Connector by the average size you expect your VMs to be, then add about 10GB for the virtual appliance. For example, if you project that you will run a maximum of 5 imported VMs averaging 75GB in size, then you'll need about 385GB of disk space.

**Internet access**

- Outbound Internet access, either direct or via a proxy, from the Connector appliance on TCP port 443 (SSL) to Amazon EC2 and Amazon S3.

**Note**

If you are adding a firewall rule to allow this access and want to further restrict this access to Amazon EC2 and Amazon S3, you can add the hosts at the published endpoints as the destinations on TCP port 443. For more information about the current endpoints, see [Regions and Endpoints](#).

- DHCP server
- A static IP address or an IP reservation via DHCP for the virtual appliance

**Note**

You must set up a static DHCP lease or configure a static IP before you begin the configuration process.

### Connector local network access

- Inbound TCP 443 (SSL) from vCenter Server and vSphere Client
- Inbound TCP 80 (HTTP) from the LAN for Connector Web Console

### Administrative rights

- To VMware vSphere and VMware vCenter for installation. For information on how to allow a user without administrative rights to use the Connector to import VMs to Amazon EC2, see [To grant permission to non-administrative users to import VMs to Amazon EC2 \(p. 318\)](#).

### AWS access credentials

- The Connector stores AWS credentials for each VMware vCenter user. In this way, multiple users with separate AWS credentials can use the same Connector. Alternatively, you can use the same AWS account and credentials with more than one user. Each VMware vSphere user will need to fill out the information in the **Enter AWS Credentials** dialog box the first time he or she uses the Connector. For information on how to get your AWS credentials, see [Get Your Access Key ID and Secret Access Key \(p. 561\)](#).
- Your AWS account must be subscribed to Amazon EC2 before you start the import process.

**Note**

The Connector virtual appliance stores your AWS credentials. To protect these credentials from unauthorized use, grant access to the virtual appliance's console and configuration only to administrators.

## Installing the Connector for VMware vCenter

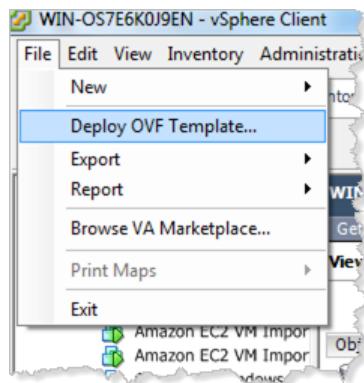
After you have confirmed that you have all the prerequisites and your virtualization environment meets the minimum requirements, you are now ready to install the Connector virtual appliance (vApp) for VMware vCenter.

The Connector virtual appliance is an Open Virtualization Format (OVF) package that is distributed in an Open Virtualization Application (OVA) file. Installing the Connector involves downloading the OVA file and deploying the OVF template.

### To Install the Connector for the VMware vCenter

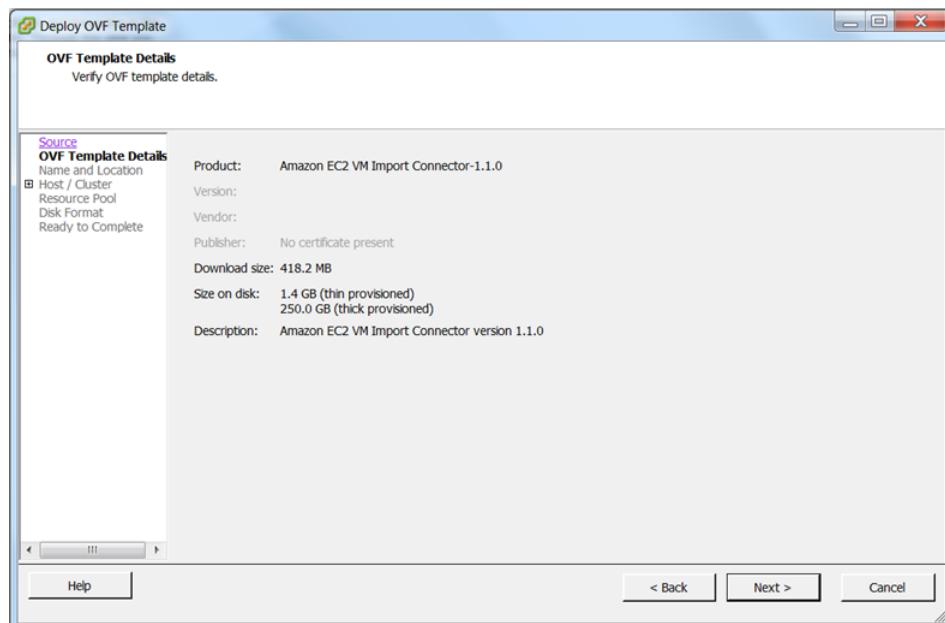
1. Download the OVA package for the Connector virtual appliance from [Amazon Web Services Developer Tools](#) and save it to your Downloads folder.
2. Start the vSphere Client and connect to your vCenter Server.

3. Using the vSphere Client, deploy the OVF template contained in the OVA file that you downloaded. On the **File** menu, select **Deploy OVF Template** and point to the location where you downloaded the OVA package.

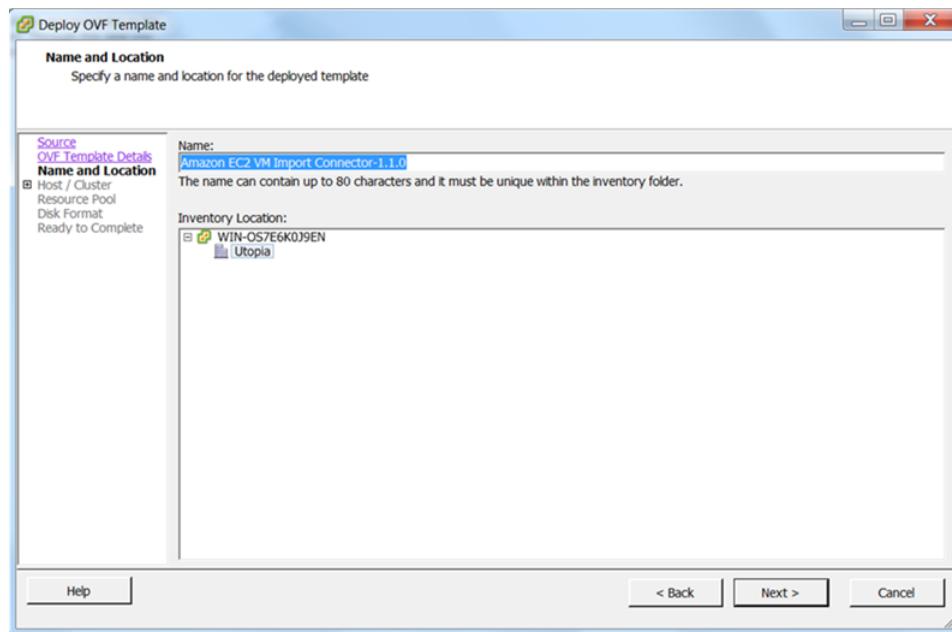


A series of **Deploy OVF Template** screens walks you through the deployment process.

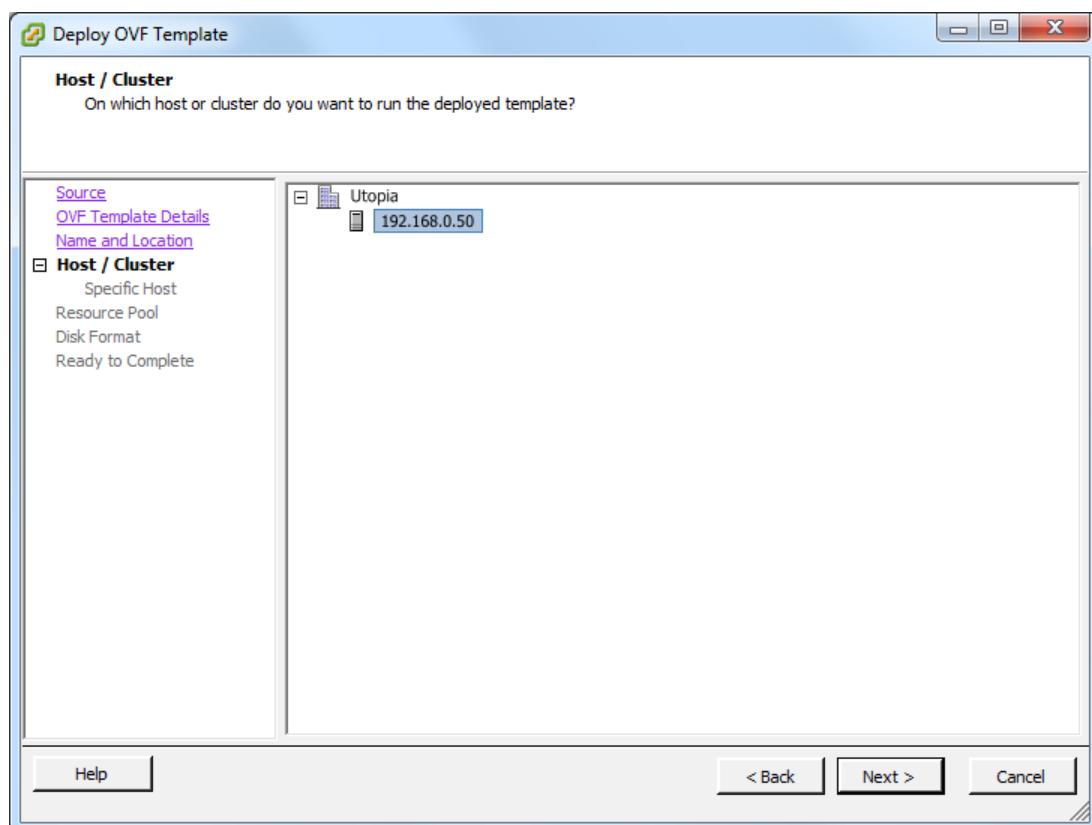
4. Confirm that the **Deploy OVF Template** screen is displaying correct information about the VM Import Connector, and click **Next**.



5. Specify the name and location of the Connector or accept the default, then click **Next**.



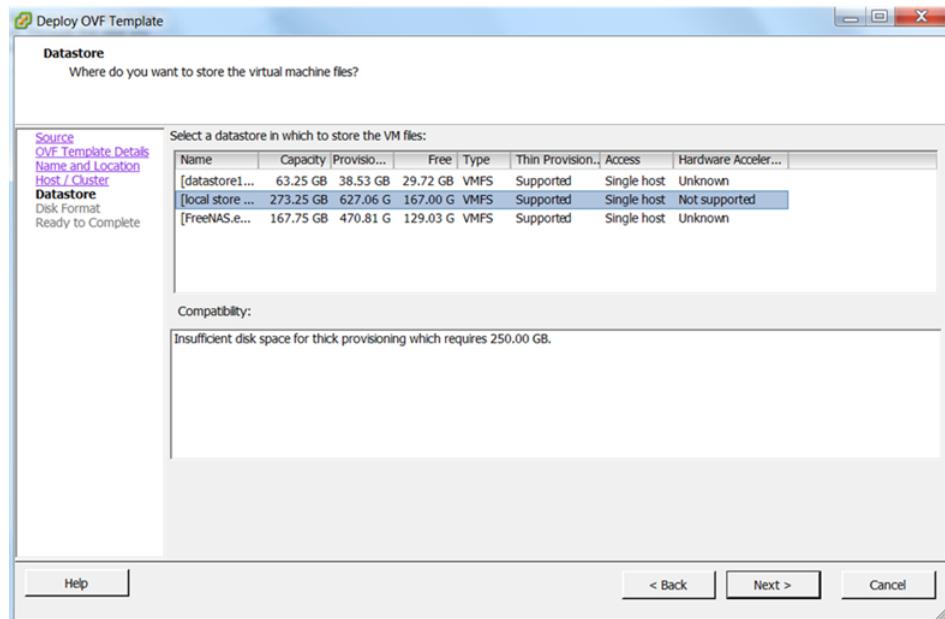
6. Select the host or cluster in which you want the Connector to run, then click **Next**.



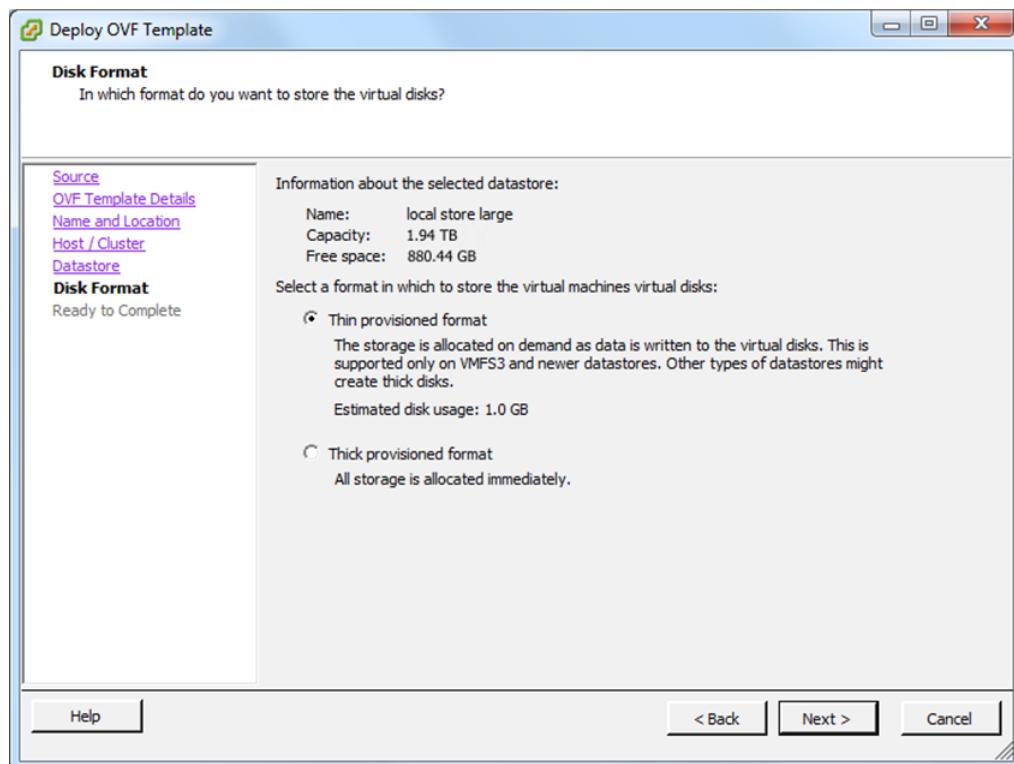
7. Select the data store where the Connector will be stored, then click **Next**.

**Note**

Although the Connector virtual appliance is small, it temporarily stores the images of the virtual machines that import to Amazon EC2. Therefore, the data store must be large enough to accommodate these images, as well as the Connector. We recommend a data store size of 250GB or larger.



8. Select **Thin provisioned format**, then click **Next**.



9. Confirm the details you selected and click **Finish**.

The Connector virtual appliance will now install.

## Configuring the Connector for VMware vCenter

After you install the EC2 VM Import Connector, you must obtain its IP address and password and register it with the vCenter Server. To obtain the Connector's IP address and password, you first start the Connector appliance in vCenter, then go to the vCenter **Console** tab. The tab displays the IP address and password when the Connector is running. In a web browser, use this information to log in to the Connector and register it with the vCenter Server.

If the VMware vSphere Client was running when you installed the Connector virtual application, close and restart the vSphere Client, then follow these procedures.

### To start the Connector for VMware vCenter

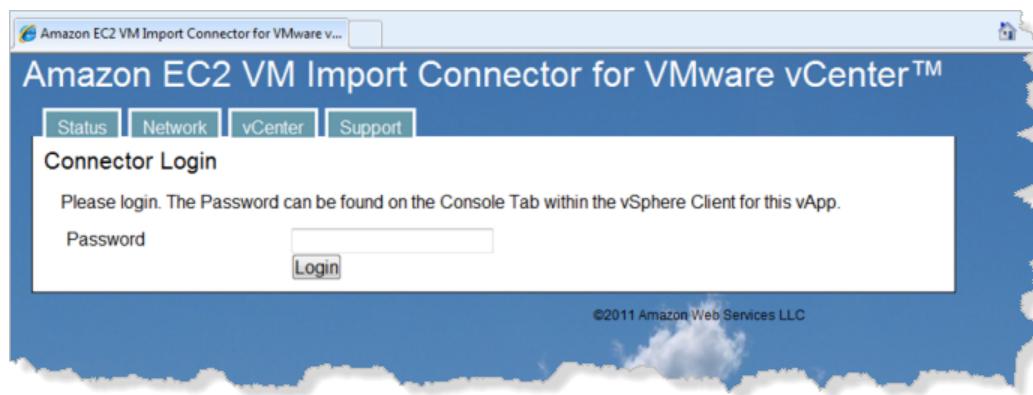
1. On the vSphere Client, right-click the Connector you just installed, select **Power** and then **Power On**.
2. To open the console, right-click the Connector you just installed and click **Open Console**.

When the Connector is running, you will find its IP address and password displayed in the vCenter **Console**. Your Connector information will be similar to the following example:

```
Amazon EC2 VM Import Connector for VMware vCenter
Management Website -> 192.0.2.99
Management Password -> tVRILx3K
login: ■
```

### To register the Connector with the VMware vCenter

1. Open a web browser and, in the address bar, type the Connector IP address that you obtained from the **Console** in the vCenter, and log in with the password.



2. When you are logged in, the browser displays Connector status information. Note that the Connector is not yet registered with the vCenter Server. Confirm that everything else in the **Connector Status** list has a status of **OK**.



3. If the version of the Connector that you are registering is not an upgrade, click **Register Now**. If you are upgrading the Connector, you must take the next two steps before registering.
  - Confirm that all import tasks are complete.
  - Deregister the old Connector from vCenter. To do this, go to the **vCenter** tab and click **Deregister**.



The **vCenter Connector Registration** page appears.

4. Provide the IP address of the vCenter with which you want to register the Connector. The user name and password you use must have administrative rights. Click **Register Connector with vCenter**.

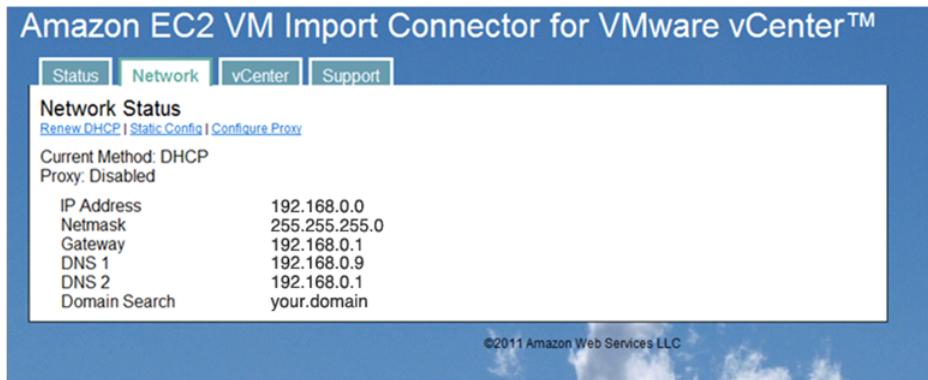
**Note**

If an error occurs, check that the VMware vCenter IP address or name that you provided is correct. Also, confirm that the Connector virtual appliance has network access to TCP port 443 on your vCenter Server. If the user name or password you provided is incorrect, you will see a description of this error.

### To configure a proxy

If you need to configure a proxy to allow the Connector to reach the Internet, you can do it using the Connector's web interface.

1. Go to the **Network** tab, click **Configure Proxy**.



The **Network Configuration:Proxy** page appears.

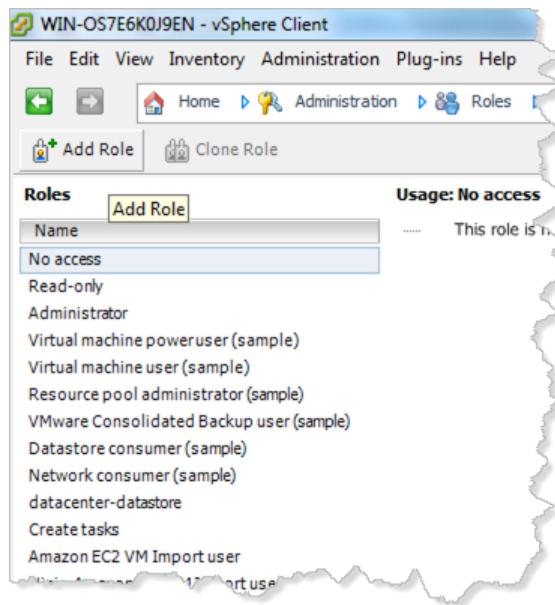
2. Provide the information required and click **Configure Proxy**.



After using the web browser to register the Connector with the vCenter Server, as an administrator you can import virtual machines to Amazon EC2. If you're going to import VMs using only an administrator account, skip the following section and go to [Using the Connector for VMware vCenter \(p. 321\)](#). If you want non-administrative users to import VMs to Amazon EC2, you must grant them permission using the vCenter.

#### To grant permission to non-administrative users to import VMs to Amazon EC2

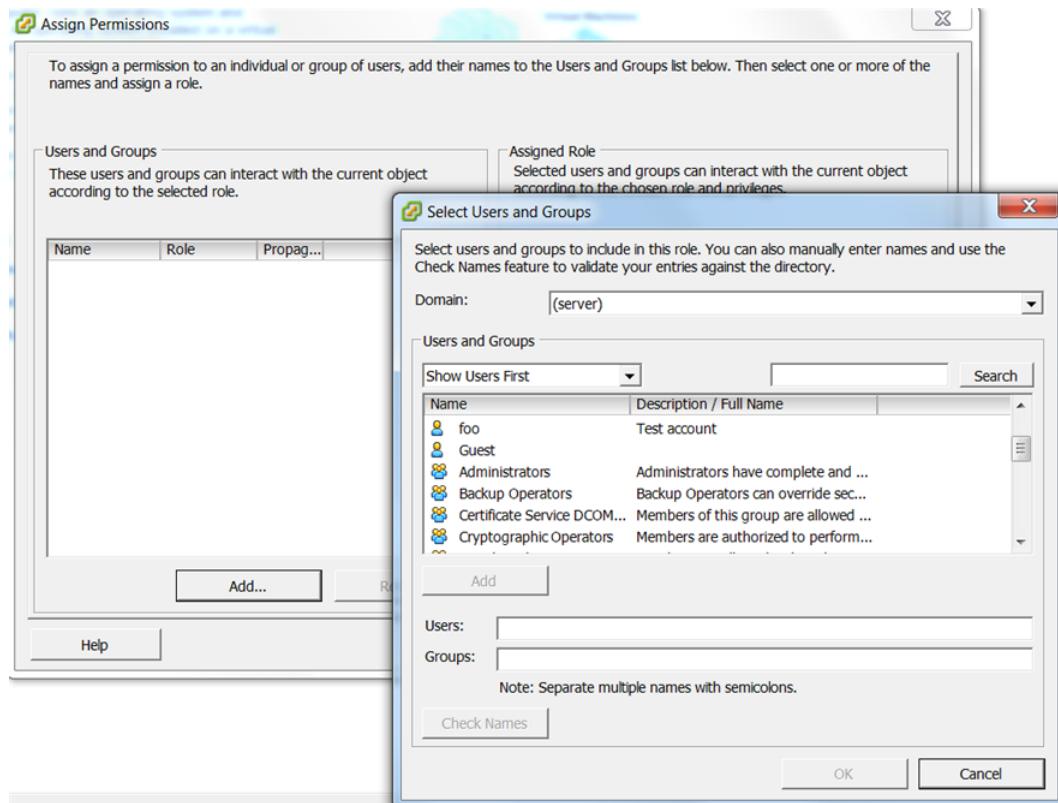
1. Log in to vCenter as Administrator and from **Home**, navigate to **Roles**, and click **Add Role**.



2. In the **Add New Role** dialog box, type the name for the new role and specify the following permissions.
  - Under **Global**, select **Cancel task**.
  - Under **Tasks**, select **Create task** and **Update task**.
  - Under **vApp**, select **Export** and **View OVF Environment**.

Click **OK**.

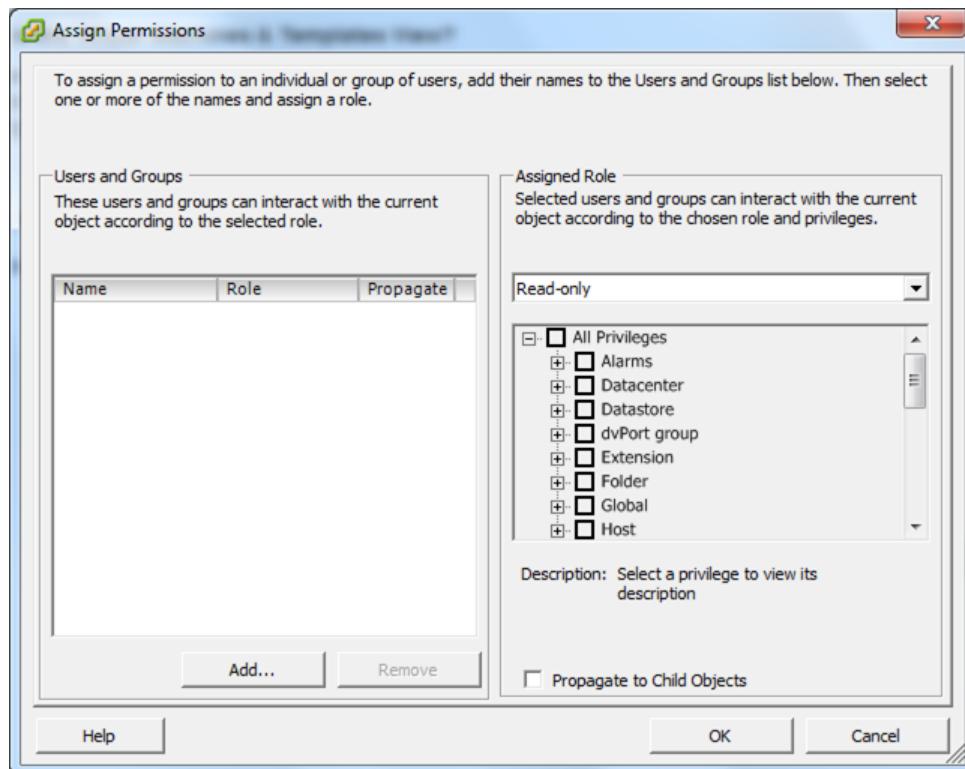
3. To grant the new role permission to vCenter users or groups who will be importing virtual machines to Amazon EC2, right-click your specific vCenter in the tree-view pane and select **Add permission**. The **Assign Permissions** dialog box opens.
4. In the **Users and Groups** box on the left, select the users or groups you want to add to the new role you created.



If you don't have users defined yet, click **Add**. The **Select Users and Groups** dialog box opens.

5. Select and add the users you want to add to the new role. When you have identified all the users for the role, click **OK**.
6. In the **Assigned Role** box on the right of the **Assign Permissions** dialog box, select the role that you previously created.
7. Clear the check box for **Propagate to Child Objects** and click **OK**.

Repeat the same process to assign the role of virtual machine power user to all users and groups that you want to allow to import VMs to Amazon EC2. You can do this at the VM object level or at a higher level in the hierarchy. If you assign the role at a higher level, you must select the check box for **Propagate to Child Objects**.



## Using the Connector for VMware vCenter

This section shows you how to use the Connector to import a virtual machine to Amazon EC2 for the first time using an account with administrative rights.

Confirm that you have prepared the virtual machine according to the guidelines in [Preparing Your Virtual Machine \(p. 310\)](#).

### Important

If you don't enable RDP and disable the Windows-based firewall, your VM will import successfully to Amazon EC2, but you will not be able to log in.

These requirements must be satisfied:

- The virtual machine must be turned off.
- The virtual machine must only use a single virtual hard drive (multiple partitions are OK).
- The virtual hard drive cannot be larger than one terabyte (1TB).
- The Connector virtual appliance must have sufficient free hard drive space to temporarily store the compressed VMDK image while it is being imported to Amazon EC2.

### To use the Connector to import a VM for the first time using an account with administrative rights

1. Log in to VMware vCenter using the VMware vSphere Client. If you had a session open while you were installing the Connector, notice that the **Import to EC2** tab becomes visible.

### Note

The first time you log in, you will see an SSL certificate warning. This warning indicates that the SSL certificate being used by the Connector cannot be verified by an external source.

This is expected behavior. Your session will continue to use SSL for encryption. Check the **Install this certificate and do not display a security warning** option at the bottom of the screen and click **Ignore**.

You are now logged in to the vCenter Server.

2. On the left pane, navigate the tree view to the virtual machine you want to import. Select it.
3. On the right pane, select the **Import to EC2** tab.

**Note**

You might see another security warning about SSL certificates. Click **Yes** to continue.

The Connector initializes.

4. In the **Enter AWS Credentials** dialog box, provide your **Access Key ID** and **Secret Access Key**. For information on how to get your AWS credentials, see [Get Your Access Key ID and Secret Access Key \(p. 561\)](#).

The main Connector page appears when your AWS credentials are verified.

5. Back in vCenter, select the virtual machine you want to import and go to the **Import to EC2** tab.



6. In the **Instance Parameters** dialog box, specify the values for the following options, then click the **Import to EC2** button.

- **Region**—(Required) The AWS Region in which you want to launch your imported instance. Select one of the options from the drop-down list.
- **Availability Zone**—(Optional) Your choice of Availability Zone in the Region that you have selected. If not specified, Amazon will select one for you.
- **Operating System**—(Required) The operating system that your virtual machine is using. Selecting an incorrect value can cause the import task to fail.

For virtual machines using Windows Server 2008 R2, always select **Microsoft Windows Server 2008 (64-bit)**. If the virtual machine you're importing runs on Windows Server 2008 SP2, determine first whether the operating system is a 32-bit or 64-bit **System Type** then select the **Operating System** accordingly. For more information about 32-bit and 64-bit Windows, go to [32-bit and 64-bit Windows: frequently asked questions](#). For more information about how to determine System Type for Windows Server 2003, go to [Microsoft Support](#).

- **Instance Type**—(Required) The instance type to use when your virtual machine becomes an instance in EC2. Instance types allow you to choose the CPU and memory requirements. For more information on instance types and pricing, go to [Amazon EC2 Pricing](#).
- **Launch in VPC**—(Optional) Check this box if you want your virtual machine to become an instance within Amazon Virtual Private Cloud (Amazon VPC). For information, go to [Amazon Virtual Private Cloud \(Amazon VPC\)](#).

- **Subnet**—(Required only if launching in VPC) Select the subnet that you want your instance placed within in a VPC.
- **Private IP address**—(Optional, applies only if launching in VPC) Specify the private IP address of your instance within VPC.
- **Security Group**—(Required) Select the security group to use with your instance. Defaults to the default security group.

The values you specified are listed in the **Confirm Import Options** box. Check the information and click **Import**.

7. Monitor the progress of the import task in the **Tasks & Events** or **Recent Tasks** tab of the VMware vSphere Client.

Name	Target	Status	Details	Initiated by	vCenter Server	Requested Start Time	Start Time
Export OVFtemplate	Win 2008 Serv...	4%	In Progress	Administrator	WIN-057E8K0...	2/24/2011 1:25:45 PM	2/24/2011 1:25:45 PM
Import to EC2	Win 2008 Serv...	Completed		Administrator	WIN-057E8K0...	2/24/2011 1:24:56 PM	2/24/2011 1:25:44 PM
Export OVFtemplate	Win 2008 Serv...	Completed		Administrator	WIN-057E8K0...	2/18/2011 7:01:24 AM	2/18/2011 7:01:24 AM
Import to EC2	Win 2008 Serv...	Completed		Administrator	WIN-057E8K0...	2/18/2011 7:01:14 AM	2/18/2011 7:01:23 AM

Name	Target	Status	Details	Initiated by	vCenter Server	Requested Start Time	Start Time	Completed Time
Export OVFtemplate	Win 2008 Serv...	4%	In Progress	Administrator	WIN-057E8K0...	2/24/2011 1:25:45 PM	2/24/2011 1:25:45 PM	
Import to EC2	Win 2008 Serv...	In Progress	Waiting to...	Administrator	WIN-057E8K0...	2/24/2011 1:24:56 PM	2/24/2011 1:25:44 PM	

- **Export OVF template**—Creates a stream-optimized VMDK image. This process consolidates your virtual machine to a single image. In addition, stream-optimized VMDKs are compressed and are well-suited for transfer over a WAN connection. The stream-optimized VMDK will be temporarily stored on your Connector virtual appliance.
- **Import to EC2**—Transfers the stream-optimized VMDK that was created in the first task to Amazon EC2, and converts your virtual machine to an Amazon EC2 instance.

The **Import to EC2** task can take up to a few hours to complete. In addition, you might notice that the task progress will pause for up to 10 minutes at times. This is expected behavior.

**Important**

Keep your session in VMware vCenter open until all tasks complete. If you quit the vSphere Client or log off of your vCenter, the import task will not complete successfully, and the progress indicator will not be updated. If this occurs, you can use the command line tools to check the status of your import task. For more information, see Checking on the Status of Your Import in [Using the Command Line Tools to Import Your Virtual Machine to Amazon EC2 \(p. 324\)](#). When you have verified that the task is complete, you can safely cancel the import task by right-clicking the task in the vSphere Client and selecting **Cancel**.

Although you can see your instance in the AWS Management Console when the import process begins, do not launch your instance until the import process completes. For information about launching instances, see [Launching an Instance from an AMI \(p. 284\)](#).

## Uninstalling the Connector for VMware vCenter

If you no longer want to use the Connector for VMware vCenter and you want to uninstall the virtual appliance, you will follow a two-part process:

- Deregister the Connector from the vCenter Server.

- Shut down the virtual appliance.

#### To Uninstall the Connector from the VMware vCenter

1. Open a web connection to the Connector's IP address and log in.
2. Click the **vCenter** tab, then click **Deregister**.
3. In the **vCenter Connector Deregistration** page, enter the vCenter IP information and user name and password, then click **Deregister Connector with vCenter**.
4. When the Connector is no longer registered with the vCenter Server, shut down the virtual appliance and remove it from vCenter.

## Using the Command Line Tools to Import Your Virtual Machine to Amazon EC2

### Topics

- [Exporting Your Virtual Machines from Their Virtual Environment \(p. 325\)](#)
- [Importing Your Virtual Machine into Amazon EC2 \(p. 330\)](#)

In this section, you'll learn how to use the Amazon EC2 command line tools to import your Citrix, Microsoft Hyper-V, or VMware virtual machine to Amazon EC2. If you haven't already installed the Amazon EC2 command line tools, see [Setting Up the Amazon EC2 Command Line Interface Tools on Linux/UNIX \(p. 560\)](#).

Importing VMs into Amazon EC2 is a two-step process. First, you export your virtual machine from the virtualization environment. Next, you create an import task and upload your virtual machine into Amazon EC2.

Use the following commands when you perform import tasks using the Amazon EC2 command line tools:

Command	Description
<a href="#">ec2-import-instance</a>	Creates a new import instance task using metadata from the specified disk image and imports the instance to Amazon EC2.
<a href="#">ec2-import-volume</a>	Creates a new import volume task using metadata from the specified disk image and imports the volume to Amazon EC2.
<a href="#">ec2-resume-import</a>	Resumes the upload of a disk image associated with an import instance or import volume task ID.
<a href="#">ec2-describe-conversion-tasks</a>	Lists and describes your conversion tasks.
<a href="#">ec2-cancel-conversion-task</a>	Cancels the active conversion task. The task can be the import of an instance or volume.
<a href="#">ec2-delete-disk-image</a>	Deletes a partially or fully uploaded disk image for conversion from Amazon S3.

For information about these commands and other EC2 commands, go to the [Amazon Elastic Compute Cloud Command Line Reference](#).

**Note**

You can use a graphical user interface provided through the Amazon EC2 VM Import Connector (Connector) to import your VMware virtual machines to Amazon EC2. For more information, see [Using the Amazon EC2 VM Import Connector to Import Your Virtual Machine to Amazon EC2 \(p. 310\)](#). You cannot use the Connector to import Citrix or Microsoft Hyper-V virtual machines.

## Exporting Your Virtual Machines from Their Virtual Environment

The process of exporting virtual machines depends on the source virtualization environment. In this section, we will show you the basic steps to export virtual machines from Citrix, Microsoft Hyper-V, and VMware virtualization products. For in-depth information, consult the documentation for these products.

- [Exporting from Citrix \(p. 325\)](#)
- [Exporting from Microsoft Hyper-V \(p. 328\)](#)
- [Exporting from VMware \(p. 329\)](#)

Before starting the export process, prepare the Windows Server environment of the virtual machine you are exporting so that:

- Remote desktop is enabled.
- Windows firewall allows public RDP traffic.
- Autologon is disabled.
- There are no pending Microsoft updates and the computer is not set to install software when it reboots.

### Exporting from Citrix

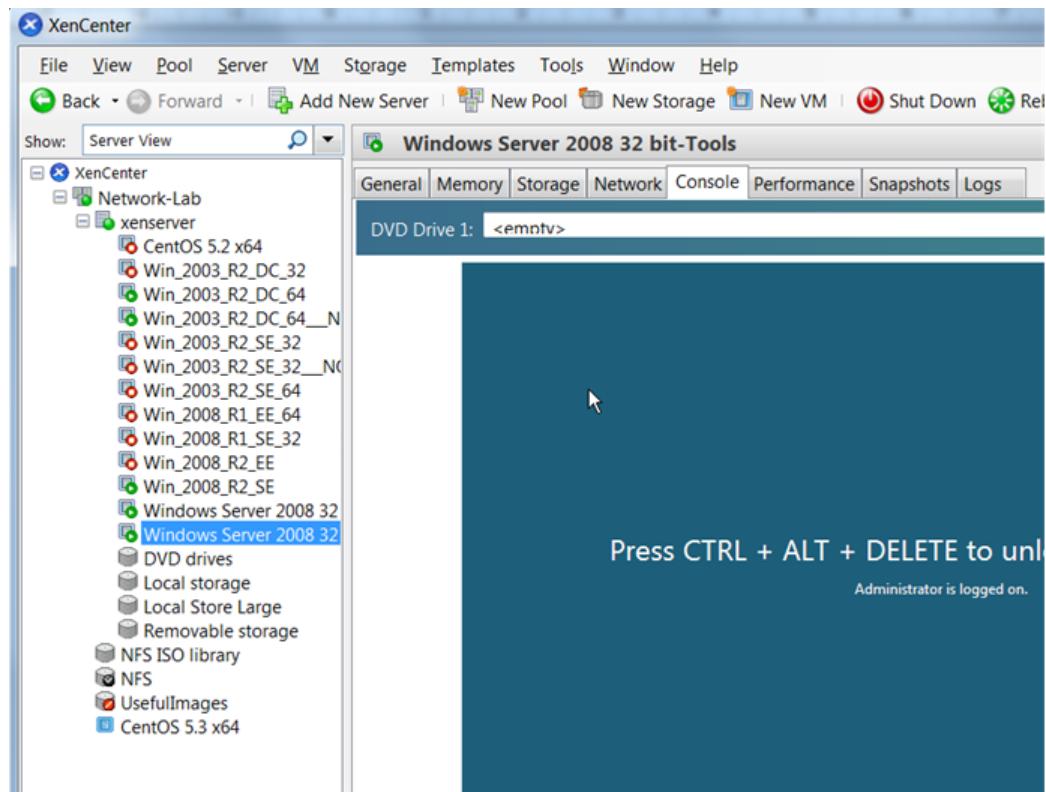
Before you export your virtual machines from Citrix XenCenter, you must perform the following tasks:

- Remove the **Citrix Tools for Virtual Machines** from the VM.
- Use the Citrix XenCenter console to remove the tools.

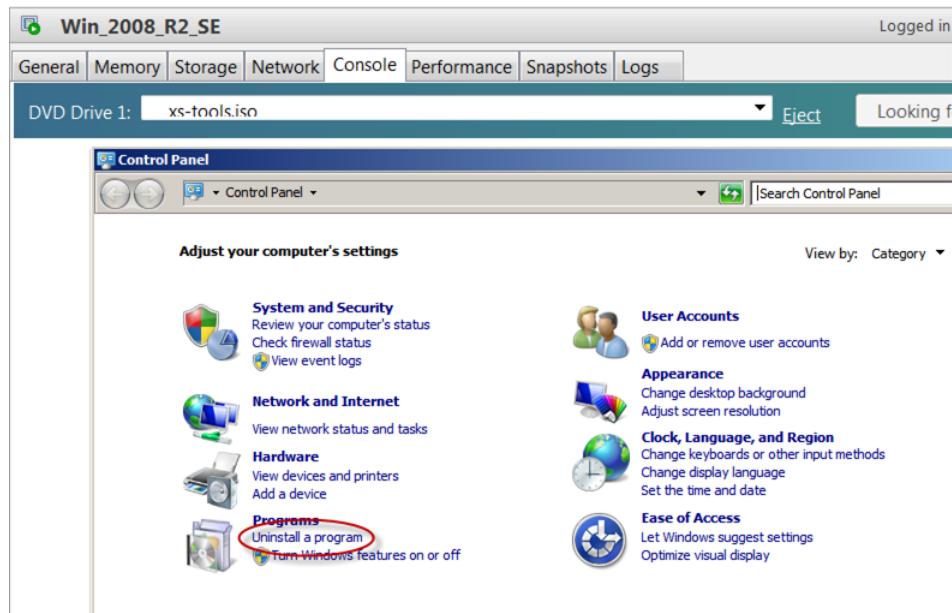
If you have multiple virtual disks that you want to export from Citrix Xen, we recommend that you export the disks one at a time. The export of multiple virtual disks at the same time results in a list of randomly named VHD files.

#### To remove Citrix tools for virtual machines

1. In **XenCenter**, select the virtual machine you want to export, and click the **Console** tab, which shows the Windows desktop of the virtual machine.



- Using the **Console**, access the Control Panel of the virtual machine's Windows operating system and uninstall the **Citrix Tools for Virtual Machines**.



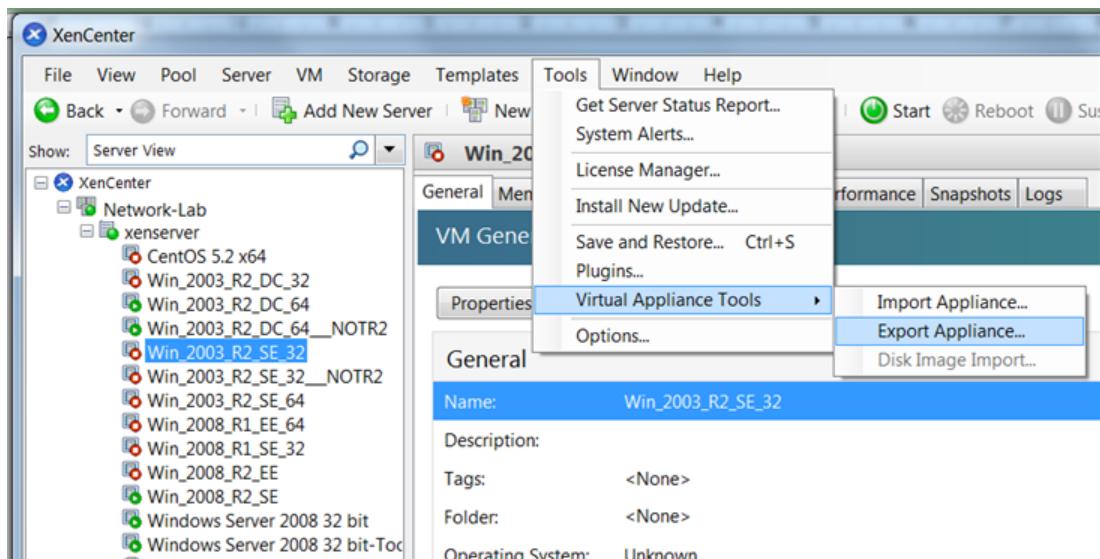
- After the tools are removed, reboot the virtual machine when prompted, log in again and then shut down using Windows.  
You can now proceed to export the VM.

### To export an image from Citrix

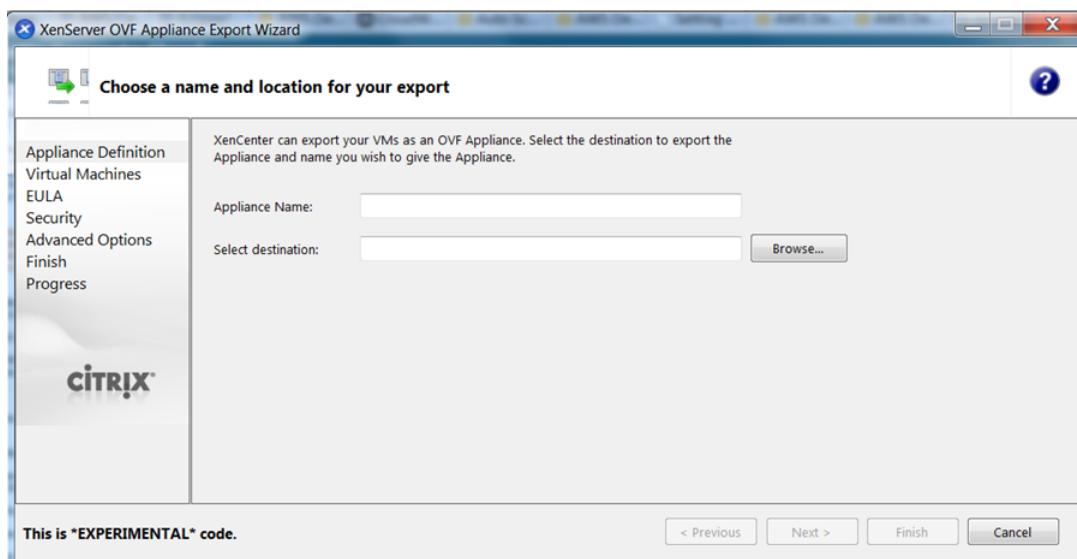
1. In the Citrix XenCenter, select the virtual machine you want to export, and then go to the **Tools** menu, click **Virtual Appliance Tools**, and then **Export Appliance**.

**Note**

Do not export the image by right-clicking a stopped instance and selecting "Export to File."



2. When the **XenServer OVF Appliance Export Wizard** starts, specify the destination of the VM files, click **Next**, accept the defaults, and proceed through the screens until you click **Finish**.



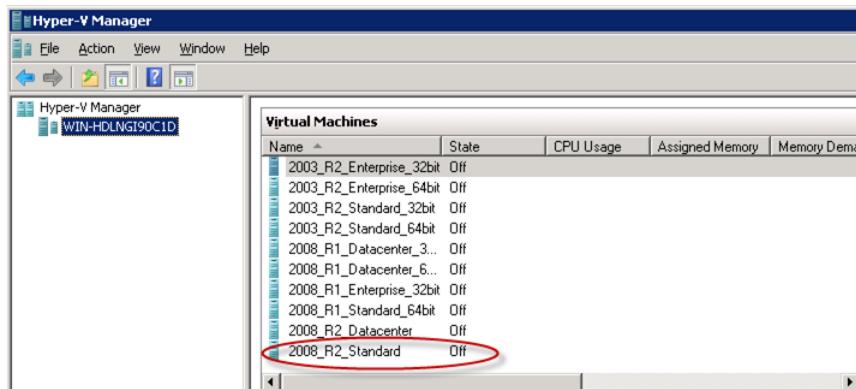
3. When the export completes, you can proceed and import the VM files to Amazon EC2 by following the steps in [Importing Your Virtual Machine into Amazon EC2 \(p. 330\)](#) and specifying VHD as the file format.

## Exporting from Microsoft Hyper-V

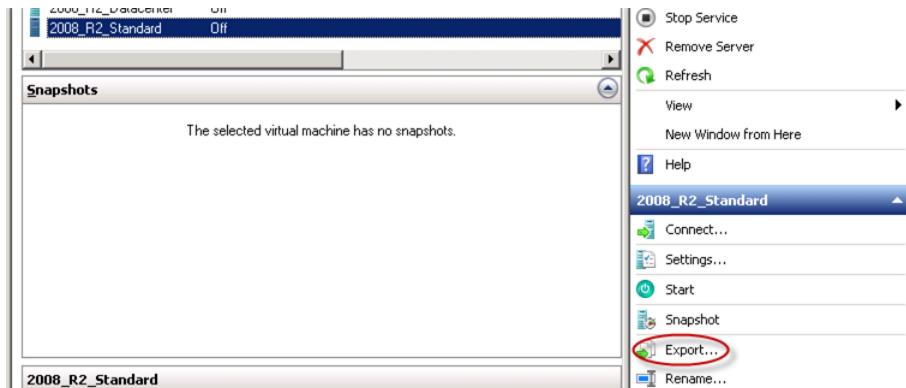
To export Hyper-V virtual disks from Microsoft, you use the Hyper-V Manager.

### To export a Hyper-V image from Microsoft

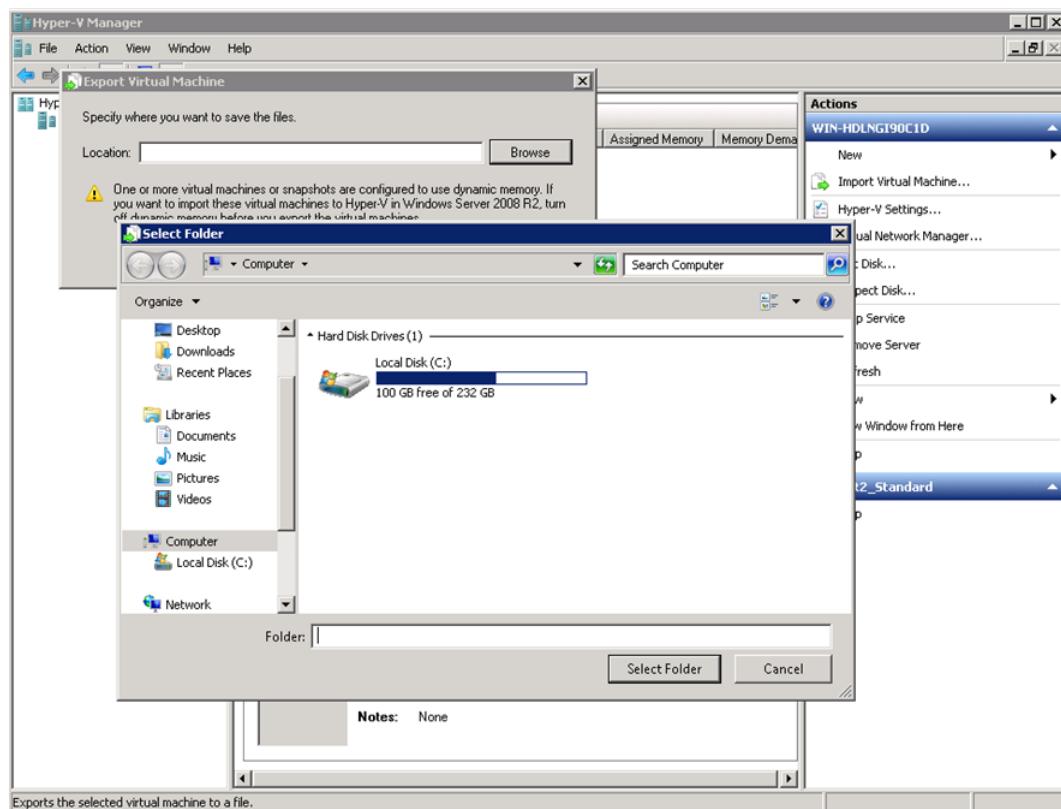
1. In the **Hyper-V Manager**, shut down the virtual machine you want to export.



2. In the **Actions** pane for the virtual machine, select **Export**.



3. In the **Export Virtual Machine** dialog box, for **Location**, click **Browse**, navigate to a destination location that has plenty of space, and click **Export**.



4. Track the export progress through the **Status** of your VM in the Hyper-V Manager. Wait for the export to complete.

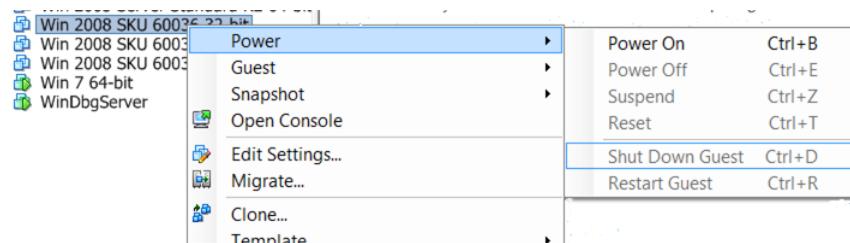
When the export completes, you can proceed and import the VM files to Amazon EC2 by following the steps in [Importing Your Virtual Machine into Amazon EC2 \(p. 330\)](#) and specifying VHD as the file format. The VHD file will be located in the folder you specified in the **Export Virtual Machine** dialog box.

## Exporting from VMware

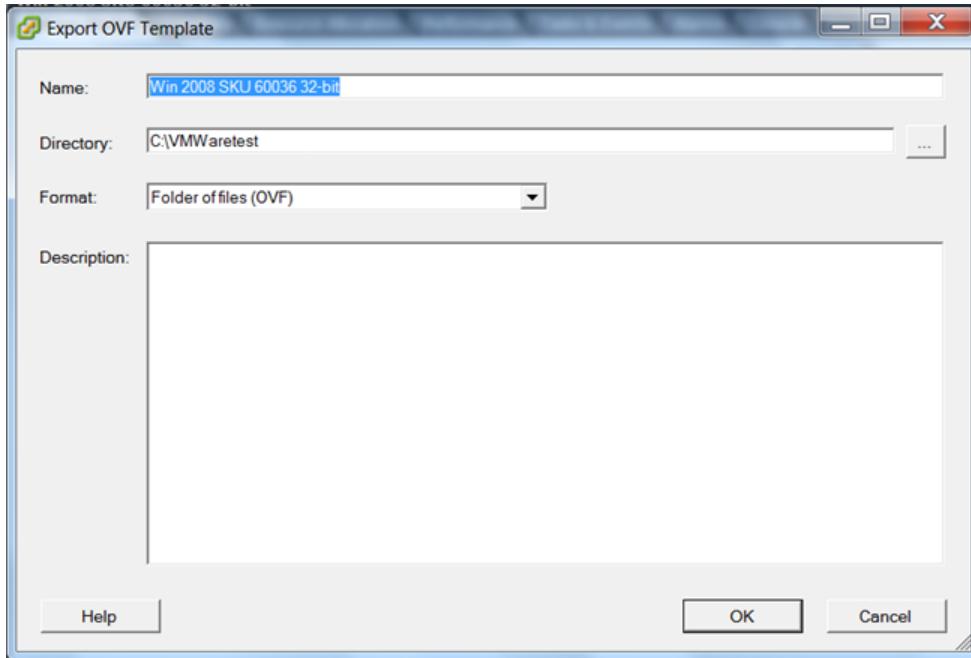
Before you can import a VMware vSphere VM or volume into Amazon EC2, you must export the VMDK disk image file. The following procedure shows you how to use the VMware vSphere Client to export a VM (and VMDK file). For more detailed information, consult your VMware documentation.

### To export a disk image from VMware

1. In the VMware vSphere Client, select the virtual machine to export.
2. Shut down the machine.



3. From the **File** menu, select **Export**, and then **Export OVF Template**.



4. In the **Export OVF Template** dialog box, enter a name for the disk image file and a directory to save it in.



You will see a box displaying the progress of the export. Because of the size and number of files associated with the VM, and the network connection, the export might take some time. If the connection times out, the export process will fail. If this happens, restart the export. When the export completes, vSphere saves the disk image file in the directory you specified. Use the name of the VMDK file as an argument in `ec2-import-instance` to import a virtual machine into Amazon EC2, or in `ec2-import-volume` to import a volume into Amazon Elastic Block Store (Amazon EBS).

## Importing Your Virtual Machine into Amazon EC2

After exporting your virtual machine from the third-party virtualization environment, you can import it into Amazon EC2. The import process is the same regardless of the origin of the virtual machine.

Here are some important things to know about your VM instance, as well as some security and storage recommendations:

- Amazon EC2 automatically assigns a DHCP IP address to your instance. The DNS name and IP address are available via the `ec2-describe-instances` command when the instance starts running.
- Your instance has only one Ethernet network interface.

- To specify an Amazon Virtual Private Cloud (Amazon VPC) subnet to use when you create the conversion task, use the `--subnet subnet_id` option with the `ec2-import-instance` command. Otherwise your instance will use a public IP address. We recommend you use a restrictive security group to control access to your instance. For more information about the `--subnet subnet_id` option, see [ec2-import-instance](#).
- We recommend that your instance contain strong passwords for all user accounts.
- We recommend that you install the [Amazon Windows EC2Config Service](#) after you import your virtual machine into Amazon EC2.

### To import a virtual machine

You can import a virtual machine into Amazon EC2. If the import of the virtual machine is interrupted, you can use the `ec2-resume-import` command to resume the import from where it stopped. For more information, see [Resuming an Upload \(p. 334\)](#).

- Use `ec2-import-instance` to create a new import instance task.  
The syntax of the command is:

```
ec2-import-instance DISK_IMAGE_FILENAME -t INSTANCETYPE -f FORMAT -a ARCHITECTURE-SYSTEM -b S3_BUCKET_NAME -o OWNER -w SECRETKEY
```

The following command creates an import instance task that imports a Windows Server 2008 SP2 (32-bit) VM.

#### Note

The example uses the VMDK format. You can also use VHD or RAW.

```
ec2-import-instance ./WinSvr8-2-32-disk1.vmdk -f VMDK -t m1.small -a i386
-b myawsbucket -o AKIAIOSFODNN7EXAMPLE -w wJalrXUtnFEMI/K7MDENG/bPxRfi
CYEXAMPLEKEY
```

This request uses the VMDK file, `WinSvr8-2-32-disk1.vmdk`, to create the import task. The output is similar to the following example.

```
Requesting volume size: 25 GB
Disk image format: Stream-optimized VMDK
Converted volume size: 26843545600 bytes (25.00 GiB)
Requested EBS volume size: 26843545600 bytes (25.00 GiB)
TaskType IMPORTINSTANCE TaskId import-i-fhbx6hua ExpirationTime
 2011-09-09T15:03:38+00:00 Status active StatusMessage Pending
InstanceID i-6ced060c
DISKIMAGE DiskImageFormat VMDK DiskImageSize 5070303744
VolumeSize 25 AvailabilityZone us-east-1c Approximate
BytesConverted 0 Status active StatusMessage Pending
Creating new manifest at testImport/9cba4345-b73e-4469-8106-
2756a9f5a077/Win_2008_R1_EE_64.vmdkmanifest.xml
Uploading the manifest file
Uploading 5070303744 bytes across 484 parts
0% |-----| 100%
|=====|
Done
```

## Checking on the Status of Your Import

The `ec2-describe-conversion-tasks` command returns the status of an import. Status values include:

- **active**—Your instance or volume is still importing.
- **cancelling**—Your instance or volume is still being canceled.
- **cancelled**—Your instance or volume is canceled.
- **completed**—Your instance or volume is ready to use.

The imported instance is in the stopped state. You use `ec2-start-instance` to start it. For more information, go to [ec2-start-instances](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

### To check the status of your import

- Use `ec2-describe-conversion-task` to return the status of the task. The syntax of the command is:

```
ec2-describe-conversion-tasks TASKID
```

The following example enables you to see the status of your import instance task.

```
ec2-describe-conversion-tasks import-i-ffvko9js
```

The following response shows that the `IMPORTINSTANCE` status is `active`, and 73747456 bytes out of 893968896 have been converted.

```
TaskType IMPORTINSTANCE TaskId import-i-ffvko9js ExpirationTime
2011-06-07T13:30:50+00:00 Status active StatusMessage Pending
InstanceID i-17912579
DISKIMAGE DiskImageFormat VMDK DiskImageSize 893968896 VolumeSize
12 AvailabilityZone us-east-1a ApproximateBytesCon
verted 73747456 Status active StatusMessage Pending
```

The following response shows that the `IMPORTINSTANCE` status is `active`, and at 7% progress and that the `DISKIMAGE` is completed.

```
TaskType IMPORTINSTANCE TaskId import-i-ffvko9js ExpirationTime
2011-06-07T13:30:50+00:00 Status active StatusMessage Progress:
7% InstanceID i-17912579
DISKIMAGE DiskImageFormat VMDK DiskImageSize 893968896 VolumeId
vol-9b59daf0 VolumeSize 12 AvailabilityZone us-
east-1a ApproximateBytesConverted 893968896 Status completed
```

The following response shows that the `IMPORTINSTANCE` status is `completed`.

```
TaskType IMPORTINSTANCE TaskId import-i-ffvko9js ExpirationTime
2011-06-07T13:30:50+00:00 Status completed InstanceID i-
17912579
DISKIMAGE DiskImageFormat VMDK DiskImageSize 893968896 VolumeId
vol-9b59daf0 VolumeSize 12 AvailabilityZone us-
east-1a ApproximateBytesConverted 893968896 Status completed
```

**Note**

The IMPORTINSTANCE status is what you use to determine the final status. The DISKIMAGE status will be completed for a period of time before the IMPORTINSTANCE status is completed.

You can now use commands such as `ec2-stop-instance`, `ec2-start-instance`, `ec2-reboot-instance`, and `ec2-terminate-instance` to manage your instance.

**Note**

By default, when you terminate an instance, Amazon EC2 does not delete the associated Amazon EBS volume. You can optionally use the `ec2-modify-instance-attribute` command to change this behavior.

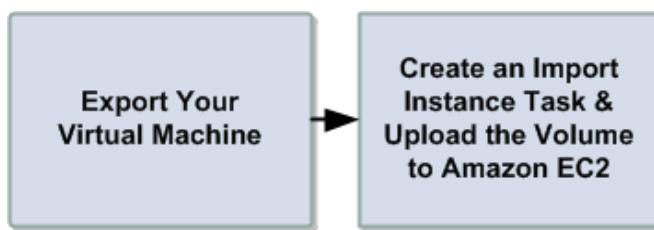
## Importing Your Volumes into Amazon EBS

This section describes how to import your data storage into Amazon EBS, and then attach it to one of your existing Amazon EC2 instances. Amazon EC2 supports importing RAW and VMDK disk formats.

**Important**

We recommend utilizing Amazon EC2 security groups to limit network access to your imported instance. Configure a security group to allow only trusted Amazon EC2 instances and remote hosts to connect to RDP and other service ports. For more information about security groups, see [Amazon EC2 Security Groups \(p. 415\)](#).

After you have exported your virtual machine from the virtualization environment, importing the volume to Amazon EBS is a single-step process. You create an import task and upload the volume in one step, as illustrated in the following diagram.



### To import a volume into Amazon EBS

1. Use `ec2-import-volume` to create a task that allows you to upload your volume into Amazon EBS. The syntax of the command is:

```
$ ec2-import-volume DISK_IMAGE_FILENAME -f FORMAT -s SIZE-IN-GB -z AVAILABILITY_ZONE -b S3_BUCKET_NAME -o OWNER -w SECRETKEY
```

The following example creates an import volume task for importing a volume to the us-east-1 region.

**Note**

The example uses the VMDK format. You can also use VHD or RAW.

```
Requesting volume size: 25 GB
Disk image format: Stream-optimized VMDK
Converted volume size: 26843545600 bytes (25.00 GiB)
Requested EBS volume size: 26843545600 bytes (25.00 GiB)
TaskType IMPORTVOLUME TaskId import-vol-ffut5xv4 ExpirationTime
 2011-09-09T15:22:30+00:00 Status active StatusMessage Pending
DISKIMAGE DiskImageFormat VMDK DiskImageSize 5070303744
```

```
VolumeSize 25 AvailabilityZone us-east-1d Approximate
BytesConverted 0
Creating new manifest at tesdfgting/0fd8fcf5-04d8-44ae-981f-
3c9f56d04520/Win_2008_R1_EE_64.vmdkmanifest.xml
Uploading the manifest file
Uploading 5070303744 bytes across 484 parts
0% |-----| 100%
|=====|
Done
```

Amazon EC2 returns a task ID that you use in the next step. In this example, the ID is import-vol-ffut5xv4.

2. Use `ec2-describe-conversion-tasks` to confirm that your volume imported successfully.

```
$ ec2-describe-conversion-tasks import-vol-ffut5xv4
TaskType IMPORTVOLUME TaskId import-vol-ffut5xv4 ExpirationTime
 2011-09-09T15:22:30+00:00 Status completed
DISKIMAGE DiskImageFormat VMDK DiskImageSize 5070303744
VolumeId vol-365a385c VolumeSize 25 AvailabilityZone
 us-east-1d ApproximateBytesConverted 5070303744
```

The status in this example is completed, which means the import succeeded.

3. Use `ec2-attach-volume` to attach the Amazon EBS volume to one of your existing Amazon EC2 instances. The following example attaches the volume, `vol-2540994c`, to the `i-a149ec4a` instance on the device, `/dev/sde`.

```
$ ec2-attach-volume vol-2540994c -i i-a149ec4a -d /dev/sde
ATTACHMENT vol-2540994c i-a149ec4a /dev/sde attaching 2010-03-
23T15:43:46+00:00
```

## Resuming an Upload

Connectivity problems can interrupt an upload. When you resume an upload, Amazon EC2 automatically starts the upload from where it stopped. The following procedure steps you through determining how much of an upload succeeded and how to resume it.

### To resume an upload

- Use the task ID with `ec2-resume-import` to continue the upload. The command uses the HTTP HEAD action to determine where to resume.

```
ec2-resume-import DISK_IMAGE_FILENAME -t TASK_ID -o OWNER -w SECRETKEY
```

The following example resumes an import instance task.

```
Disk image size: 5070303744 bytes (4.72 GiB)
Disk image format: Stream-optimized VMDK
Converted volume size: 26843545600 bytes (25.00 GiB)
Requested EBS volume size: 26843545600 bytes (25.00 GiB)
Uploading 5070303744 bytes across 484 parts
0% |-----| 100%
|=====|
```

```
Done
Average speed was 10.316 MBps
The disk image for import-i-ffni8aei has been uploaded to Amazon S3
where it is being converted into an EC2 instance. You may monitor the
progress of this task by running ec2-describe-conversion-tasks. When
the task is completed, you may use ec2-delete-disk-image to remove the
image from S3.
```

## Canceling an Upload

Use `ec2-cancel-conversion-task` to cancel an active conversion task. The task can be the upload of an instance or a volume. The command removes all artifacts of the conversion, including uploaded volumes or instances.

If the conversion is complete or still transferring the final disk image, the command fails and returns an exception similar to the following:

```
Client.CancelConversionTask Error: Failed to cancel conversion task import-i-fh95npoc
```

### To cancel a conversion task

- Use the task ID of the upload you want to delete with `ec2-cancel-conversion-task`. The following example cancels the upload associated with the task ID `import-i-fh95npoc`.

```
PROMPT> ec2-cancel-conversion-task import-i-fh95npoc
```

The output for a successful cancellation is similar to the following:

```
CONVERSION-TASK import-i-fh95npoc
```

You can use the `ec2-describe-conversion-tasks` command to check the status of the cancellation. For example:

```
$./ec2-describe-conversion-tasks import-i-fh95npoc
TaskType IMPORTINSTANCE TaskId import-i-fh95npoc ExpirationTime
2010-12-20T18:36:39+00:00 Status cancelled InstanceID i-825063ef
DISKIMAGE DiskImageFormat VMDK DiskImageSize 2671981568
VolumeSize 40 AvailabilityZone us-east-1c ApproximateBytesCon
verted 0 Status cancelled
```

In the above example, the status is `cancelled`. If it were still in process, the status would be `cancelling`.

## Cleaning up After an Upload

You can use `ec2-delete-disk-image` to remove the image file after it is uploaded. If you do not delete it, you will be charged for its storage in Amazon S3.

### To delete a disk image

- Use the task ID of the disk image you want to delete with `ec2-delete-disk-image`.

The following example deletes the disk image associated with the task ID, `import-i-fh95npoc`.

```
PROMPT> ec2-delete-disk-image import-i-fh95npoc
```

The output for a successful cancellation is similar to the following:

```
DELETE-TASK import-i-fh95npoc
```

## Troubleshooting Instance Importation

When you import a virtual machine using the `ec2-import-instance` command, the import task might stop at 56 percent completion, and then fail. To investigate what went wrong, you can use the `ec2-describe-conversion-tasks` command to describe the instance. Then, you should see the following message:

*FirstBootFailure: This import request failed because the Windows instance failed to boot and establish network connectivity.*

When you receive this message it means that your virtual disk image was unable to perform one of the following steps:

- Boot up and start Windows.
- Install Amazon EC2 networking and disk drivers.
- Use a DHCP-configured network interface to retrieve an IP address.
- Activate Windows using the Amazon EC2 Windows volume license.

## Best Practices for Avoiding First Boot Failures

The following best practices can help you to avoid Windows first boot failures.

### Disable anti-virus and anti-spyware software and firewalls.

These types of software can prevent installing new Windows services or drivers or prevent unknown binaries from running. Software and firewalls can be re-enabled after importing.

### Do not harden your operating system.

Security configurations, sometimes called hardening, can prevent unattended installation of EC2 drivers. There are numerous Windows configuration settings that can prevent import. These settings can be reapplied once imported.

### Disable or delete multiple bootable partitions.

If your virtual machine boots and requires you to choose which boot partition to use, the import may fail.

## Possible Causes of First Boot Failures

This inability of the virtual disk image to boot up and establish network connectivity could be due to any of the following causes.

### Topics

- [The installation of Windows is not valid on the virtual machine. \(p. 337\)](#)
- [TCP/IP networking and DHCP are enabled. \(p. 337\)](#)

- A volume that Windows requires is missing from the virtual machine. (p. 337)
- Windows always boots into System Recovery Options. (p. 337)
- The virtual machine was created using a physical-to-virtual (P2V) conversion process. (p. 338)
- Windows Activation fails. (p. 338)

### The installation of Windows is not valid on the virtual machine.

#### Cause

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

#### Resolution

Do not run SYSPREP before shutting down the Amazon EC2 instance. After the instance is imported, you can run SYSPREP from the instance before you create an Amazon Machine Image (AMI). Importing creates a single instance, so running SYSPREP is not necessary.

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

### TCP/IP networking and DHCP are enabled.

#### Cause

For any Amazon EC2 instance, including those in Amazon VPC, TCP/IP networking and DHCP must be enabled. Within a VPC you can define an IP address for the instance either before or after importing the instance. Do not set a static IP address before exporting the instance.

#### Resolution

Ensure that TCP/IP networking is enabled. For more information, see [Setting up TCP/IP \(Windows Server 2003\)](#) or [Configuring TCP/IP \(Windows Server 2008\)](#) at the *Microsoft TechNet website*.

Ensure that DHCP is enabled. For more information, see [What is DHCP](#) at the *Microsoft TechNet web site*.

### A volume that Windows requires is missing from the virtual machine.

#### Cause

Importing a virtual machine into Amazon EC2 only imports the boot disk, all other disks must be detached and Windows must be able to boot before importing the virtual machine. For example, Active Directory often stores the Active Directory database on the D:\ drive. A domain controller cannot boot if the Active Directory database is missing or inaccessible.

#### Resolution

Detach any secondary and network disks attached to the Windows virtual machine before exporting.

Move any Active Directory databases from secondary drives or partitions onto the primary Windows partition. For more information, see ["Directory Services cannot start" error message when you start your Windows-based or SBS-based domain controller](#) at the *Microsoft Support website*.

### Windows always boots into System Recovery Options.

#### Cause

Windows can boot into System Recovery Options for a variety of reasons, including when Windows is pulled into a virtualized environment from a physical machine, also known as P2V.

### Resolution

Ensure that Windows boots to a login prompt before exporting and preparing for import.

Do not import virtualized Windows instances that have come from a physical machine.

### The virtual machine was created using a physical-to-virtual (P2V) conversion process.

#### Cause

A P2V conversion occurs when a disk image is created by performing the Windows installation process on a physical machine and then importing a copy of that Windows installation into a virtual machine. Virtual machines, which are created as the result of a P2V conversion are not supported by Amazon EC2 VM import. Amazon EC2 VM import only supports Windows images that were natively installed inside the source virtual machine.

### Resolution

Install Windows in a virtualized environment and migrate your installed software to that new virtual machine.

### Windows Activation fails.

#### Cause

During boot, Windows will detect a change of hardware and attempt activation. During the import process we attempt to switch the licensing mechanism in Windows to a volume license provided by Amazon Web Services. However, if the Windows activation process does not succeed, then the import will not succeed.

### Resolution

Ensure that the version of Windows you are importing supports volume licensing. Beta or preview versions of Windows may not.

## Exporting EC2 Instances

### Topics

- [Before You Get Started \(p. 338\)](#)
- [Export an Instance \(p. 339\)](#)
- [Cancel or Stop the Export of an Instance \(p. 340\)](#)

If you have previously imported an instance running Microsoft Windows Server into Amazon Elastic Compute Cloud (Amazon EC2), you can use the command line tools to export that Microsoft Windows Server instance to Citrix Xen, Microsoft Hyper-V, or VMware vSphere. Exporting an instance can be useful when you want to deploy a copy of your EC2 instance in your on-site virtualization environment.

## Before You Get Started

Before you begin the process of exporting an instance, you need to be aware of the operating systems and image formats we support, and understand the limitations on exporting instances and volumes. You will also need to download and install the EC2 command line tools and sign up for your private key and X.509 certificate before you use the command line interface (CLI) or the API to export your instance. For more information, see [Setting Up the Amazon EC2 Command Line Interface Tools on Linux/UNIX \(p. 560\)](#).

## Operating Systems

The following operating systems can be exported from Amazon EC2:

- Windows Server 2003 R2 (Standard, Enterprise, and Datacenter)
- Windows Server 2008 (Standard, Enterprise, and Datacenter)
- Windows Server 2008 R2 (Standard, Enterprise, and Datacenter)

## Image Formats

We support the following image formats for exporting both volumes and instances from Amazon Web Services (AWS):

- Stream-optimized ESX Virtual Machine Disk (VMDK) image format, which is compatible with VMware ESX and VMware vSphere versions 4 and 5 virtualization products.
- Open Virtual Appliance (OVA) image format, which is compatible with VMware vSphere versions 4 and 5.
- Virtual Hard Disk (VHD) image format, which is compatible with Citrix Xen and Microsoft Hyper-V virtualization products.

## Known Limitations

The exporting of instances and volumes is subject to the following limitations:

- You cannot export Amazon Elastic Block Store (Amazon EBS) data volumes.
- You cannot export an instance that has more than one virtual disk.
- You cannot export an instance that has more than one network interface.

## Export an Instance

You can use the Amazon EC2 command line interface (CLI) to export an instance. The [ec2-create-instance-export-task](#) command gathers all of the information necessary (e.g., instance ID; name of the S3 bucket that will hold the exported image; name of the exported image; VMDK, OVA, or VHD format) to properly export the instance to the selected virtualization format. The exported file is saved in the Amazon Simple Storage Service (Amazon S3) bucket that you designate.

### Note

When you export an instance, you are charged the standard Amazon S3 rates for the bucket where the exported VM is stored. In addition, a small charge reflecting temporary use of an EBS snapshot might appear on your bill. For more information about Amazon S3 pricing, see [Amazon Simple Storage Service \(S3\) Pricing](#).

### To export an instance

1. Create an Amazon S3 bucket where exported instances will be stored. The S3 bucket must grant **Upload/Delete** and **View Permissions** access to the [vm-import-export@amazon.com](mailto:vm-import-export@amazon.com) account. For more information, see [Creating a Bucket](#) and [Editing Bucket Permissions](#) in the Amazon Simple Storage Service Console User Guide.
2. At a command prompt, type the following command: `ec2-create-instance-export-task INSTANCE_ID -e TARGET_ENVIRONMENT -f DISK_IMAGE_FORMAT -c CONTAINER_FORMAT -b S3_BUCKET`

Where:

*INSTANCE\_ID* is the ID of the instance you want to export.

*TARGET\_ENVIRONMENT* is VMware, Citrix, or Microsoft.

*DISK\_IMAGE\_FORMAT* is VMDK for VMware or VHD for Microsoft Hyper-V and Citrix Xen.

*CONTAINER\_FORMAT* may be optionally set to OVA when exporting to VMware.

*S3\_BUCKET* is the name of the Amazon S3 bucket to which you want to export the instance.

3. To monitor the export of your instance, at the command prompt, type the following command:  
`ec2-describe-export-tasks TASK_ID`

Where:

*TASK\_ID* is the ID of the export task.

## Cancel or Stop the Export of an Instance

You can use the Amazon EC2 command line interface (CLI) to cancel or stop the export of an instance up to the point of completion. The `ec2-cancel-export-task` command removes all artifacts of the export, including any partially created Amazon S3 objects. If the export task is complete or is in the process of transferring the final disk image, the command fails and returns an error.

### To cancel or stop the export of an instance

- At the command prompt, type: `ec2-cancel-export-task TASK_ID`

Where

*TASK\_ID* is the ID of the export task you want to cancel.

# Managing Instances

## Topics

- [Auto Scaling and Load Balancing Your Instances \(p. 340\)](#)
- [Monitoring Your Instances \(p. 342\)](#)
- [Ensuring Idempotency \(p. 363\)](#)
- [Instance Metadata and User Data \(p. 365\)](#)

This section describes features that help you manage your instances.

## Auto Scaling and Load Balancing Your Instances

If you expect your application to have significant variability in usage, you might want to use Auto Scaling and Elastic Load Balancing, two features of Amazon EC2 that help manage variability.

### Auto Scaling

Auto Scaling enables you to scale the number of instances that you are using up or down, 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 Auto Scaling groups, which enable you to scale different server classes (for example, 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 information about setting up Auto Scaling, see the [Auto Scaling Developer Guide](#).

## Load Balancing

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

Customers reach your web site through 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 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 information about setting up Elastic Load Balancing, see the [Elastic Load Balancing Developer Guide](#).

# Monitoring Your Instances

## Topics

- [Monitoring Your Instances with CloudWatch \(p. 343\)](#)
- [Monitoring the Status of Your Instances \(p. 354\)](#)

Amazon Web Services (AWS) automatically provides data, such as Amazon CloudWatch metrics and instance status, that you can use to monitor your Amazon EC2 instances:

- CloudWatch metrics are statistical data you can use to view, analyze, and set alarms on the operational behavior of your instances. These metrics include CPU utilization, network traffic, I/O, and latency.
- Instance status provides two types of information:
  - Instance status checks that summarize results of automated tests that you can use to determine whether your instances are affected by specific, detectable problems.
  - Events that provide information about certain activities that are scheduled for your instances, including operational maintenance that AWS may perform such as rebooting and retirement.

For information about monitoring your Amazon EBS volumes, see [Monitoring the Status of Your Volumes \(p. 478\)](#).

## Monitoring Your Instances with CloudWatch

### Topics

- [Monitoring Instances \(p. 343\)](#)
- [Creating and Editing Status Check Alarms \(p. 351\)](#)

Amazon CloudWatch is a service that collects raw data from partnered AWS products 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. For detailed information about Amazon CloudWatch, see the [Amazon CloudWatch Developer Guide](#).

### Monitoring Instances

The following table describes the types of monitoring data available for your Amazon EC2 instances.

Resource	Type	Description
Instances	Basic	Data is available automatically in 5-minute periods at no charge.
Detailed	<p>Data is available in 1-minute periods at an additional cost. To get this level of data, you must specifically enable it for the instance. For the instances where you've enabled detailed monitoring, you can also get aggregated data across groups of similar instances.</p> <p>For information about pricing, go to the <a href="#">Amazon CloudWatch product page</a>.</p>	

You can get monitoring data for your Amazon EC2 instances using either the Amazon CloudWatch API or the AWS Management Console. The console displays a series of graphs based on the raw data from the Amazon CloudWatch API. Depending on your needs, you might prefer to use either the data from the API or the graphs in the console.

### Data from the Amazon Cloudwatch API

You can use the Amazon CloudWatch `GetMetricStatistics` API action to get any of the instance metrics listed in the following table. The *period* refers to how often the system reports a data point for each metric for an instance. If you've enabled detailed monitoring, each data point covers the instance's previous 1 minute of activity. Otherwise, each data point covers the instance's previous 5 minutes of activity.

Metric	Description
CPUUtilization	<p>The percentage of allocated EC2 compute units that are currently in use on the instance. This metric identifies the processing power required to run an application upon a selected instance.</p> <p>Units: <i>Percent</i></p>
DiskReadOps	<p>Completed read operations from all ephemeral disks available to the instance (if your instance uses Amazon EBS, see <a href="#">Amazon EBS Metrics (p. 478)</a>.)</p> <p>This metric identifies the rate at which an application reads a disk. This can be used to determine the speed in which an application reads data from a hard disk.</p> <p>Units: <i>Count</i></p>
DiskWriteOps	<p>Completed write operations to all ephemeral disks available to the instance (if your instance uses Amazon EBS, see <a href="#">Amazon EBS Metrics (p. 478)</a>.)</p> <p>This metric identifies the rate at which an application writes to a hard disk. This can be used to determine the speed in which an application saves data to a hard disk.</p> <p>Units: <i>Count</i></p>
DiskReadBytes	<p>Bytes read from all ephemeral disks available to the instance (if your instance uses Amazon EBS, see <a href="#">Amazon EBS Metrics (p. 478)</a>.)</p> <p>This metric is used to determine the volume of the data the application reads from the hard disk of the instance. This can be used to determine the speed of the application.</p> <p>Units: <i>Bytes</i></p>
DiskWriteBytes	<p>Bytes written to all ephemeral disks available to the instance (if your instance uses Amazon EBS, see <a href="#">Amazon EBS Metrics (p. 478)</a>.)</p> <p>This metric is used to determine the volume of the data the application writes onto the hard disk of the instance. This can be used to determine the speed of the application.</p> <p>Units: <i>Bytes</i></p>
NetworkIn	<p>The number of bytes received on all network interfaces by the instance. This metric identifies the volume of incoming network traffic to an application on a single instance.</p> <p>Units: <i>Bytes</i></p>
NetworkOut	<p>The number of bytes sent out on all network interfaces by the instance. This metric identifies the volume of outgoing network traffic to an application on a single instance.</p> <p>Units: <i>Bytes</i></p>

Metric	Description
StatusCheckFailed	<p>A combination of StatusCheckFailed_Instance and StatusCheckFailed_System that reports if either of the status checks has failed. Values for this metric are either 0 (zero) or 1 (one.) A zero indicates that the status check passed. A one indicates a status check failure.</p> <p><b>Note</b></p> <p>Status check metrics are available at 5 minute frequency and are not available in Detailed Monitoring. For a newly launched instance, status check metric data will only be available after the instance has completed the initialization state. Status check metrics will become available within a few minutes of being in the running state.</p> <p>Units: <i>Count</i></p>
StatusCheckFailed_Instance	<p>Reports whether the instance has passed the EC2 instance status check in the last 5 minutes. Values for this metric are either 0 (zero) or 1 (one.) A zero indicates that the status check passed. A one indicates a status check failure.</p> <p><b>Note</b></p> <p>Status check metrics are available at 5 minute frequency and are not available in Detailed Monitoring. For a newly launched instance, status check metric data will only be available after the instance has completed the initialization state. Status check metrics will become available within a few minutes of being in the running state.</p> <p>Units: <i>Count</i></p>
StatusCheckFailed_System	<p>Reports whether the instance has passed the EC2 system status check in the last 5 minutes. Values for this metric are either 0 (zero) or 1 (one.) A zero indicates that the status check passed. A one indicates a status check failure.</p> <p><b>Note</b></p> <p>Status check metrics are available at 5 minute frequency and are not available in Detailed Monitoring. For a newly launched instance, status check metric data will only be available after the instance has completed the initialization state. Status check metrics will become available within a few minutes of being in the running state.</p> <p>Units: <i>Count</i></p>

**Note**

When you get data from Amazon CloudWatch, you can include a `Period` request parameter to specify the granularity of the returned data. This is different than the period we use when we collect the data (either 1-minute periods for detailed monitoring, or 5-minute periods for basic monitoring). We recommend that you specify a period in your request that is equal to or larger than the collection period to ensure that the returned data is valid.

You can use the dimensions in the following table to refine the metrics returned for your instances.

Dimension	Description
AutoScalingGroupName	This dimension filters the data you request for all instances in a specified capacity group. An <i>AutoScalingGroup</i> is a collection of instances you define if you're using the Auto Scaling service. This dimension is available only for EC2 metrics when the instances are in such an AutoScalingGroup. Available for instances with Detailed or Basic Monitoring enabled.
ImageId	This dimension filters the data you request for all instances running this EC2 Amazon Machine Image (AMI). Available for instances with Detailed Monitoring enabled.
InstanceId	This dimension filters the data you request for the identified instance only. This helps you pinpoint an exact instance from which to monitor data. Available for instances with Detailed Monitoring enabled.
InstanceType	This dimension filters the data you request for all instances running with this specified instance type. This helps you categorize your data by the type of instance running. For example, you might compare data from an m1.small instance and an m1.large instance to determine which has the better business value for your application. Available for instances with Detailed Monitoring enabled.

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

### Graphs in the AWS Management Console

After you launch an instance, you can go to the AWS Management Console and view the instance's monitoring graphs. They're displayed when you select the instance on the **Instances** page in the EC2 Dashboard. A **Monitoring** tab is displayed next to the instance's **Description** tab. The following graphs are available:

- Average CPU Utilization (Percent)
- Average Disk Reads (Bytes)
- Average Disk Writes (Bytes)
- Maximum Network In (Bytes)
- Maximum Network Out (Bytes)
- Summary Disk Read Operations (Count)
- Summary Disk Write Operations (Count)
- Summary Status (Any)
- Summary Status Instance (Count)
- Summary Status System (Count)

The AWS Management Console contains a console for Amazon CloudWatch. In the Amazon CloudWatch console you can search and browse all your AWS resource metrics, view graphs to troubleshoot issues and discover trends, create and edit alarms to be notified of problems, and see at-a-glance overviews of your alarms and AWS resources. For more information, see [AWS Management Console](#) in the [Amazon CloudWatch Developer Guide](#).

### Enabling or Disabling Detailed Monitoring on an Amazon EC2 Instance

This section describes how to enable or disable detailed monitoring on either a new instance (as you launch it) or on a running or stopped instance. After you enable detailed monitoring, the Amazon EC2

console in the AWS Management Console displays monitoring graphs with a 1-minute period for the instance. You can enable or disable detailed monitoring using the console or the command line interface (CLI).

## AWS Management Console

### To enable detailed monitoring of an existing EC2 instance

You can enable detailed monitoring of your EC2 instances, which provides data about your instance in 1-minute periods. (There is an additional charge for 1-minute monitoring.) Detailed data is then available for the instance in the AWS Management Console graphs or through the API. To get this level of data, you must specifically enable it for the instance. For the instances on which you've enabled detailed monitoring, you can also get aggregated data across groups of similar instances. An instance must be running or stopped to enable detailed monitoring.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the **Navigation** pane, click **Instances**.
3. In **My Instances**, select a running or stopped instance, click **Instance Actions**, and then click **Enable Detailed Monitoring**.
4. In the **Enable Detailed Monitoring** dialog box, click **Yes, Enable**.
5. In the **Enable Detailed Monitoring** confirmation dialog box, click **Close**.

Detailed data (collected with a 1-minute period) is then available for the instance in the AWS Management Console graphs or through the API.

### To enable detailed monitoring when launching an EC2 instance

- When launching an instance with the AWS Management Console, select the check box for **Enable detailed CloudWatch Monitoring for this instance** on the **Advanced Instance Options** section of the Classic launch wizard.

After the instance is launched, you can select the instance in the console and view its monitoring graphs on the instance's **Monitoring** tab in the lower pane.

### To disable detailed monitoring of an EC2 instance

When you no longer want to monitor your instances at 1-minute intervals, you can disable detailed monitoring and use basic monitoring instead. Basic monitoring provides data in 5-minute periods at no charge.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the **Navigation** pane, click **Instances**.
3. In **My Instances**, select an instance, click **Instance Actions**, and then click **Disable Detailed Monitoring**.
4. In the **Disable Detailed Monitoring** dialog box, click **Yes, Disable**.
5. In the **Disable Detailed Monitoring** confirmation dialog box, click **Close**.

For information about launching instances, see [Launching EC2 Instances \(p. 274\)](#).

## Command Line Interface

### To enable detailed monitoring on an existing instance

- Use the `ec2-monitor-instances` command with one or more instance IDs.

```
PROMPT> ec2-monitor-instances i-1a2b3c4d
i-1a2b3c4d monitoring-pending
```

Detailed data (collected with a 1-minute period) is then available for the instance in the AWS Management Console graphs or through the API.

### To enable detailed monitoring when launching an instance

- Use the `ec2-run-instances` command with the `--monitor` flag.

```
PROMPT> ec2-run-instances ami-2bb65342 -k gsg-keypair --monitor
```

Amazon EC2 returns output similar to the following example. The status of monitoring is listed as *monitoring-enabled*.

```
RESERVATION r-7430c31d 111122223333 default
INSTANCE i-ae0bf0c7 ami-2bb65342 pending gsg-keypair 0
m1.small 2008-03-21T16:19:25+0000 us-east-1a monitoring-enabled
```

After the instance is running, detailed data (collected with a 1-minute period) is then available for the instance in the AWS Management Console graphs or through the API.

### To disable detailed monitoring of an instance

- Use the `ec2-unmonitor-instances` command with one or more instance IDs.

```
PROMPT> ec2-unmonitor-instances i-1a2b3c4d
```

## API

### To enable detailed monitoring when launching an instance

- Set the `Monitoring.Enabled` parameter to `true` in the `RunInstances` request.

Following is an example Query request.

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

Following is an example response.

```
<RunInstancesResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
<reservationId>r-47a5402e</reservationId>
```

```
<ownerId>111122223333</ownerId>
<groupSet>
 <item>
 <groupId>default</groupId>
 </item>
</groupSet>
<instancesSet>
 <item>
 <instanceId>i-2ba64342</instanceId>
 <imageId>ami-60a54009</imageId>
 <instanceState>
 <code>0</code>
 <name>pending</name>
 </instanceState>
 ...
 <monitoring>
 <state>pending</state>
 </monitoring>
 ...
 </item>
</instancesSet>
</RunInstancesResponse>
```

After the instance is launched, detailed data (collected with a one-minute period) is then available for the instance in the AWS Management Console graphs or through the API.

### To enable detailed monitoring on an existing instance

- Use the `MonitorInstances` action.

Following is a sample Query request.

```
https://ec2.amazonaws.com/
?Action=MonitorInstances
&InstanceId.1=i-1a2b3c4d
&...auth parameters...
```

Following is an example response.

```
<MonitorInstancesResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">

 <instancesSet>
 <item>
 <instanceId>i-1a2b3c4d</instanceId>
 <monitoring>
 <state>pending</state>
 </monitoring>
 </item>
 </instancesSet>
</MonitorInstancesResponse>
```

Detailed data (collected with a one-minute period) is then available for the instance in the AWS Management Console graphs or through the API.

## Create a CloudWatch Alarm

You can create an Amazon CloudWatch alarm that monitors any one of your Amazon EC2 instance's CloudWatch metrics. CloudWatch will automatically send you a notification when the metric reaches a threshold you specify. You can create a CloudWatch alarm on the Amazon EC2 console or the AWS Management Console, or you can use the CloudWatch console and configure more advanced options.

### To create a CloudWatch alarm for high CPU

1. On the [Amazon EC2 console](#), select the instance for which you want to create an alarm.
2. On the instance's **Monitoring** tab in the lower pane, click **Create Alarm**.
3. In the **Create Alarm** dialog box, set the criteria for your alarm. In this example, we'll set an alarm if the instance's average CPU utilization is above 70 percent.
4. The check box next to Send a notification is selected by default. Click **Create topic**, and either use the default name, or select an existing name. (Notifications use Amazon Simple Notification Service (Amazon SNS)).
5. In the **With these recipients** box, enter the email addresses of the recipients you want to notify. You can enter up to 10 email addresses, each separated by a comma.
6. Configure the threshold for your alarm.
  - a. In the **Whenever** boxes, select **Average** and **CPU Utilization**.
  - b. In the **Is** boxes, define the threshold for the alarm by selecting **>** and entering **70**.
  - c. In the **For at least** boxes, specify the sampling period and number of samples evaluated by the alarm. You can leave the defaults or define your own. For our example, we'll monitor for 1 period of 15 minutes.

#### Note

A shorter period creates a more sensitive alarm. A longer period can mitigate brief spikes in a metric.

- d. In **Alarm name**, a name is automatically generated for you. Click **Edit** if you want to change the name.

#### Important

You cannot modify the name after you create the alarm.

**Create Alarm for MyWindowsInstance (i-33725c50)**

To create an alarm, first choose whom to notify and then define when the notification should be sent ([view all help topics](#))

**Send a notification to:**  [cancel](#)

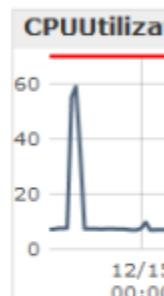
**With these recipients:**  [?](#) [cancel](#)

**Whenever:**  of  [?](#) [cancel](#)

**Is:** >"/>  Percent [?](#) [cancel](#)

**For at least:**  consecutive period(s) of  [?](#) [cancel](#)

**Name this alarm:**  [cancel](#) [?](#)



7. Click **Create Alarm**.

After you create the alarm, you can use the **Monitoring** tab in the Amazon EC2 console to view a summary of alarms that have been set for that instance. From there, you can also edit the alarm.

**Note**

If you created a new Amazon SNS topic for this alarm or added new email addresses to an existing topic, each email address added will receive a subscription confirmation email from Amazon SNS. The person who receives the email must confirm it by clicking the included link in order to receive notifications.

## Creating and Editing Status Check Alarms

You can create instance status and system status alarms to notify you when an instance has a failed status check. To create or change these alarms you can use either the AWS Management Console or the command line interface (CLI).

### AWS Management Console

#### To create a status check alarm

You can create status check alarms for an existing instance to monitor instance status or system status. You can configure the alarm to send you a notification by email when an instance fails an instance check or system status check.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the Navigation pane, click **Instances**.
3. In **My Instances**, select an instance, and then on the **Status Checks** tab, click **Create Status Check Alarm**.
4. In the **Create Alarm for** dialog box, select the **Send a notification to** check box, and then choose an existing Amazon Simple Notification Service (SNS) topic or create a new SNS topic to use for this alarm.

**Create Alarm for i-e80f7690**

You can use CloudWatch alarms to be notified automatically whenever any status check for your instance fails. To create an alarm, first choose whom to notify and then define when the notification should be sent.

**Send a notification to:**  [create topic](#)

**Whenever:**

**Is:** Failing

**For at least:**  consecutive period(s) of

**Name this alarm:**  [edit](#)

5. In the **With these recipients** box, type your email address (e.g., john.stiles@example.com) and the addresses of any additional recipients, separated by commas.
6. In the **Whenever** drop-down list, select the status check you want to be notified about (e.g., Status Check Failed (Any), Status Check Failed (Instance), or Status Check Failed (System)).
7. In the **For at least** box, set the number of periods you want to evaluate (for example, 2) and in the **consecutive periods** drop-down menu, select the evaluation period duration (for example, 5 minutes) before triggering the alarm and sending an email.
8. To change the default name for the alarm, in the **Name this alarm** box, type a friendly name for the alarm (for example, StatusCheckFailed), and then click **Create Alarm**.

**Important**

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

### To edit a status check alarm

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

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the **Navigation** pane, click **Instances**.
3. In **My Instances**, select an instance, click **Instance Actions**, and then click **Add/Edit Alarms**.
4. In the **Alarm Details** dialog box, click the name of the alarm.
5. In the **Edit Alarm for** dialog box, make the desired changes, and then click **Save Alarm**.
6. In the **Alarm Saved Successfully** dialog box, click **Close**.

## Command Line Interface

### To create a status check alarm using the CLI

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

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

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

Where:

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

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

The **--metric-name** is one of the available status metrics (e.g., StatusCheckFailed, StatusCheckFailed\_Instance, or StatusCheckFailed\_System). This is required.

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

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

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

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

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

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

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

#### Note

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

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/>.

2. On the **Navigation** pane, under **My Topics**, select the topic you want the alarm to send mail to.
3. The ARN is located in the **Topic ARN** field on the **Topic Details** pane.

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

## Monitoring the Status of Your Instances

You can monitor the status of your instances by viewing status checks and scheduled events for your instances. A status check gives you the information that results from automated checks performed by Amazon EC2. These automated checks detect whether specific issues are affecting your instances. The status check information, together with the data provided by Amazon CloudWatch, gives you detailed operational visibility into each of your instances.

You can also see status on specific events scheduled for your instances. Events provide information about upcoming activities such as rebooting or retirement that are planned for your instances, along with the scheduled start and end time of each event.

### Topics

- [Monitoring Instances with Status Checks \(p. 354\)](#)
- [Monitoring Events for Your Instances \(p. 358\)](#)

## Monitoring Instances with Status Checks

With instance status monitoring you can quickly determine whether Amazon EC2 has detected any problems that may prevent your instances from running applications. Amazon EC2 performs automated checks on every running Amazon EC2 instance to identify hardware and software issues. You can view the results of these status checks to identify specific and detectable problems. This data augments the information that Amazon EC2 already provides about the intended state of each instance (pending, running, stopping, etc.) as well as the utilization metrics that Amazon CloudWatch monitors (CPU utilization, network traffic, and disk activity).

Status checks are performed every five minutes and each returns a pass or a fail status. If all checks pass, the overall status of the instance is **OK**. If one or more checks fail, the overall status is **impaired**. Status checks are built into Amazon EC2, so they cannot be disabled or deleted. You can, however create or delete alarms that are triggered based on the result of the status checks. For example, you can create an alarm to warn you if status checks fail on a specific instance. For more information, see [Creating and Editing Status Check Alarms \(p. 351\)](#).

There are two types of status checks: system status checks and instance status checks.

**System status checks** monitor the AWS systems required to use your instance to ensure they are working properly. These checks detect problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue or you can resolve it yourself (for example, by stopping and restarting or terminating and replacing an instance). Examples of problems that cause system status checks to fail include:

- Loss of network connectivity
- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host

**Instance status checks** monitor the software and network configuration of your individual instance. These checks detect problems that require your involvement to repair. When an instance status check

fails, typically you will need to address the problem yourself (for example by rebooting the instance or by making modifications in your operating system). Examples of problems that may cause instance status checks to fail include:

- Failed system status checks
- Misconfigured networking or startup configuration
- Exhausted memory
- Corrupted file system
- Incompatible kernel

**Note**

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

## Viewing Status

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

### AWS Management Console

#### To view status checks

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the Navigation pane, click **Instances**.
3. On the **Instances** page, the **Status Checks** column lists the operational status of each instance.

Name	Instance	AMI ID	Root Device	Type	State	Status Checks	Monitoring
1 FAH n	i-34c4755e	ami-bcd53ed5	ebs	c1.medium	running	2/2 checks passed	detailed
FAH1 P	i-bd7670d7	ami-d9e40db0	ebs	m2.4xlarge	running	2/2 checks passed	detailed
2 FAH b	i-373d3b5d	ami-d9e40db0	ebs	t1.micro	running	1/2 checks passed	detailed
FAH xla	i-bd2764d0	ami-2547a34c	instance-store	c1.xlarge	running	2/2 checks passed	detailed

4. To view an individual instance's status, select the instance, and then click the **Status Checks** tab.

The screenshot shows the 'Status Checks' tab selected for an EC2 instance. A message at the top says '1 EC2 Instance selected.' Below it, the instance's URL is shown as 'EC2 Instance: [REDACTED].compute-1.amazonaws.com'. A yellow bar indicates the instance is 'Retiring' and scheduled for retirement after 2012-04-19 00:00 GMT. The 'Status Checks' tab is active, showing a table with two rows: 'System Status Checks' and 'Instance Status Checks'. The 'System Status Checks' row contains a warning icon, the text 'System reachability check failed at 2012-04-04 22:23 GMT (5 minutes, 19 seconds ago)', a 'more info' link, and a '+ Learn more about this issue' link. The 'Instance Status Checks' row also contains a warning icon, the same failure message, a 'more info' link, and a '+ Learn more about this issue' link.

### Note

If you have an instance with a failed status check and the instance has been unreachable for over 20 minutes, you can click **Contact Support** to submit a request for assistance. To try and troubleshoot system or instance status check failures yourself, see [Troubleshooting Instances with Failed Status Checks \(p. 389\)](#).

### Command Line Tools

To do this	Run this command
Get the status of all instances	<code>ec2-describe-instance-status</code>
Get the status of all instances with a instance status of impaired	<code>ec2-describe-instance-status --filter "instance-status.status=impaired"</code>
Get the status of all instances with a single instance with instance ID i-15a4417c	<code>ec2-describe-instance-status status -i-15a4417c</code>

For more information about using the **ec2-describe-instance-status** command, see [ec2-describe-instance-status](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

### Note

If you have an instance with a failed status check, see [Troubleshooting Instances with Failed Status Checks \(p. 389\)](#)

### API

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

## Reporting Status

You can provide feedback about your instances if you are having problems with an instance whose status is not shown as impaired, or to send AWS additional details about the problems you are experiencing with an impaired instance.

We use reported feedback to identify issues impacting multiple customers, but do not respond to individual account issues reported via this form. Providing feedback will not change the status check results that you currently see for this instance.

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

## AWS Management Console

### To report status feedback using the management console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the Navigation pane, click **Instances**.
3. On the **Instances** page, click on the instance on which you want to report status.
4. Click the **Status Checks** tab and then click the Submit Feedback button.

The screenshot shows the AWS Management Console interface for an EC2 instance. At the top, it says "1 EC2 Instance selected." Below that, there's a navigation bar with tabs: "Description" (disabled), "Status Checks" (selected), "Monitoring" (disabled), and "Tags" (disabled). The main content area has two sections: "System Status Checks" and "Instance Status Checks". Under "System Status Checks", there's a green checkmark next to "System reachability check passed." Under "Instance Status Checks", there's a yellow warning icon next to "Instance reachability check failed". At the bottom, there's a note: "Click Submit Feedback if our checks do not reflect your experience with this instance or if they did not detect the issues you are having:" followed by a "Submit Feedback" button.

5. Complete the information on the **Report Instance Status** page.

**Report Instance Status**

Help us improve by informing us when our checks do not reflect your experience, or when you have issues that our checks did not detect.

**Are you currently experiencing problems with this instance?**

No, there aren't any problems.  
 Yes, it's having problems.

**What problems are you experiencing? (optional)**

The instance is stuck in a state.  
 The instance is unresponsive.  
 The instance is not accepting my credentials.  
 The password is not available for this instance.  
 The instance has a network performance problem.  
 The instance has instance store performance problems.  
 The instance has an EBS volume performance problem.  
 The instance has a performance problem not listed here.  
 The instance has a different problem not listed here.

**Would you like to add any more information? (optional)**

**What happens to this feedback?**

## Command Line Tools

Use the **ec2-report-instance-status** command to send status feedback using the command line tools. The command uses the following syntax:

```
ec2-report-instance-status [instance_id ...] [--status ...] [--reason] ...
```

For more information about using the **ec2-report-instance-status** command, see [ec2-report-instance-status](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

## API

You can use the `ReportInstanceStatus` action to submit feedback about a running instance's status. If your experience with the instance differs from the instance status returned by the `DescribeInstanceStatus` action, use `ReportInstanceStatus` to report your experience with the instance. Amazon EC2 collects this information to improve the accuracy of status checks. For more information, see [ReportInstanceStatus](#) in the *Amazon Elastic Compute Cloud API Reference*.

## Monitoring Events for Your Instances

### Topics

- [Monitoring Events with Instance Status \(p. 359\)](#)
- [Working with an Instance that Has a Scheduled Event \(p. 360\)](#)

Instance status describes specific events that Amazon Web Services (AWS) may schedule for your instances. Events provide information about upcoming activities, such as rebooting or retirement, that are planned for your instances, along with the scheduled start and end time of each event. These scheduled events are not frequent. There are several types of scheduled events:

- **Instance reboot:** AWS may schedule an instance for a reboot for necessary maintenance, such as to apply patch upgrades to an underlying host that *do not* require the host to be rebooted.
- **System reboot:** AWS may schedule an instance for a reboot for necessary maintenance, such as to apply patch upgrades that *do* require the host to be rebooted.
- **Network maintenance:** AWS may schedule network maintenance that includes a scheduled start and end time, during which your instances will not have network connectivity. You will be notified by email if one of your instances is set for network maintenance. The email message indicates when your instance will not have network connectivity.
- **Power maintenance:** AWS may schedule power maintenance that includes a scheduled start and end time, during which your instances may be offline for an extended period and then be rebooted. You will be notified by email if one of your instances is set for power maintenance. The email message indicates when your instance will be rebooted.
- **Instance retirement:** AWS may schedule instances for retirement in cases where there is an unrecoverable issue with the hardware on an underlying host. You will also be notified by email if one of your instances is set to retiring. The email message indicates when your instance will be permanently retired.
- **Instance stop:** AWS may schedule instances to stop in cases where there is an unrecoverable issue with the hardware on an underlying host. You will also be notified by email if one of your instances is set to stop. The email message indicates when your instance will be stopped.

### Important

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

For instances scheduled for network maintenance, power maintenance, stop, or retirement, we recommend that you take the actions detailed later in this section.

## Monitoring Events with Instance Status

You can view scheduled events for your instances using the AWS Management Console, the command line interface (CLI), or the API.

### AWS Management Console

#### To view scheduled events for your instances

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the **Navigation** pane, click **Events**. You can see a list of all resources with events associated with them. You can view only instances list of instances that have upcoming events scheduled.
3. Alternatively, you can do the following to view upcoming scheduled events:
  - a. In the **Navigation** pane, click the **EC2 Dashboard**.
  - b. Under Events, you can see the events associated with your Amazon EC2 instances and volumes.
  - c. On the **Events** page, in **Viewing**, select **Instances** to view only instances. You can also filter on specific status types.

### Command Line Interface

#### To view scheduled events for your instances

- Enter the following command:

```
ec2-describe-instance-status
```

Amazon EC2 returns output similar to the following:

```
INSTANCE i-1a2b3c4d us-east-1d running 16 ok ok active
SYSTEMSTATUS reachability passed
INSTANCESTATUS reachability passed
INSTANCE i-2a2b3c4d us-east-1d running 16 ok ok active
SYSTEMSTATUS reachability passed
INSTANCESTATUS reachability passed
INSTANCE i-3a2b3c4d us-east-1d running 16 ok ok active
SYSTEMSTATUS reachability passed
INSTANCESTATUS reachability passed
INSTANCE i-4a2b3c4d us-east-1d running 16 ok ok retiring YYYY-MM-
DDTHH:MM:SS+0000
SYSTEMSTATUS reachability passed
INSTANCESTATUS reachability passed
EVENT instance-stop YYYY-MM-DDTHH:MM:SS+0000 The instance is running on de-
graded hardware
INSTANCE i-5a2b3c4d us-east-1d running 16 ok ok retiring YYYY-MM-
DDTHH:MM:SS+0000
SYSTEMSTATUS reachability passed
INSTANCESTATUS reachability passed
EVENT instance-retiring YYYY-MM-DDTHH:MM:SS+0000 The instance is running on de-
graded hardware
INSTANCE i-6a2b3c4d us-east-1d running 16 ok ok retiring YYYY-MM-
DDTHH:MM:SS+0000
SYSTEMSTATUS reachability passed
INSTANCESTATUS reachability passed
EVENT instance-stop YYYY-MM-DDTHH:MM:SS+0000 The instance is running on de-
graded hardware
```

For more information about using the **ec2-describe-instance-status** command, see [ec2-describe-instance-status](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

## API

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

## Working with an Instance that Has a Scheduled Event

This section discusses tasks you can perform if your instance has one of the following scheduled events.

- Instance or system reboot
- Network maintenance
- Power maintenance
- Instance retirement

### Recommended Tasks for Instances Scheduled for a Reboot

AWS may schedule instances for a reboot to perform maintenance tasks such as patch upgrades to the software on an underlying host. These scheduled reboots are not frequent. There are two types of reboot events: system reboot and instance reboot. In either case, your instance will be rebooted. During a system reboot, the hardware supporting your instance will also be rebooted. During an instance reboot, your instance will be rebooted but the hardware supporting the instance will not be rebooted.

### Important

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

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

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

### Optional Alternative Tasks for Instances Scheduled for Reboot

We recommend that you wait for the reboot to occur automatically within its scheduled window. If you choose, you may perform the instance reboot yourself to control the timing of the event.

#### Instance Reboot

For instance reboot events, you can perform the reboot yourself (be careful to not shut down or terminate your instance) at any time before the scheduled reboot event begins. The reboot can be initiated using the AWS Management Console, a `RebootInstances` API call, or from within the instance (e.g., at the command prompt). After you reboot, any pending maintenance to the underlying host is performed automatically, and you can begin using your instance again after the instance has fully booted.

#### Note

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

#### System Reboot

If you choose to perform a system reboot, your course of action will differ depending on whether your instance's root device volume is an EBS volume or an instance store volume. You can determine the root device type for an instance using either the [DescribeInstances API](#) or the AWS Management Console. In the console, you select an instance and view the root device type listed in the **Description** tab.

### Instances Backed by Amazon EBS

If your instance's root device is an EBS volume, you can stop and restart it (be careful not to shut down or terminate your instance). If you choose to do so, some configuration settings will change.

If you stop and restart your instance, the following changes occur. If you wait for Amazon EC2 to perform the scheduled reboot automatically, these configuration settings remain the same.

- Data in instance store volumes will no longer be available. Before you stop the instance, back up any data you may need. For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 513\)](#).
- The public DNS name and the private IP address of the instance will change.
- If you associated an Elastic IP address with this instance, stopping this instance also disassociates the Elastic IP address from it (charges still apply). After you start the instance, re-associate it with the Elastic IP address if the address is still required.
- For instances in Amazon Virtual Private Cloud (Amazon VPC), the Elastic IP address and the private IP address remain unchanged.

Before you stop and restart an instance, perform the following tasks:

1. Retrieve any data from instance store that you will need later. This data will not be available after you stop and restart an instance.

2. Take a snapshot of your existing volume (storage charges will apply).
3. Note the necessary configuration data in case you will need it later, including the DNS public name and the private IP address.
4. Stop and restart your instance. For more information, see [Stopping and Starting Instances \(p. 378\)](#).
5. Re-associate an Elastic IP address if this address is necessary.
6. If other applications or instances rely on the public DNS name or the private IP address of this instance, update the information with the new configuration data.

**Note**

After you stop and restart the instance, the scheduled event for the system reboot is canceled immediately. The event's description is updated in the AWS console to reflect this.

### Instances Backed by Instance Store

If your instance's root device is an instance store volume and you cannot wait for AWS to reboot your instance in the scheduled maintenance window, you can opt to launch a replacement instance. After you have launched the replacement instance, you can terminate the original instance. If you choose to take this action, be aware that:

- The public DNS name and the private IP address of your replacement instance will differ from your original instance.
- You must first ensure that you have an AMI with the configuration that you want to use when launching the new instance. For more information, see:
  - [Creating an Instance Store-Backed Windows AMI](#)
  - [Creating Instance Store-Backed Linux/UNIX AMIs \(p. 34\)](#)

### Instances Scheduled for Network Maintenance, Power Maintenance, Stop, or Retirement

The following section outlines steps you can take to migrate or replace instances that are scheduled for network maintenance, power maintenance, or retirement. If you migrate your instances scheduled for network maintenance, power maintenance, stop, or retirement before the maintenance start time, the maintenance event will automatically be cancelled.

### Instances Backed by Amazon EBS

If your instance's root device is an EBS volume, you can stop and restart it (be careful not to shut down or terminate the instance). If you choose to do so, some configuration settings will change. The following changes occur:

- Data in instance store volumes will no longer be available. Before you stop the instance, back up any data you may need. For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 513\)](#).
- The public DNS name and the private IP address of the instance will change.
- If you associated an Elastic IP address with this instance, stopping this instance also disassociates the Elastic IP address from it (charges still apply). After you start the instance, re-associate it with the Elastic IP address if the address is still required.
- For instances in Amazon Virtual Private Cloud (Amazon VPC), the Elastic IP address and the private IP address remain unchanged.

Before you stop and restart an instance, perform the following tasks:

1. Retrieve any data from instance store that you will need later. This data will not be available after you stop and restart an instance.
2. Take a snapshot of your existing volume (storage charges will apply).
3. Note the necessary configuration data in case you will need it later, including the DNS public name and the private IP address.
4. Stop and restart your instance. For more information, see [Stopping and Starting Instances \(p. 378\)](#).
5. Re-associate an Elastic IP address, if this address is necessary.
6. If other applications or instances rely on the public DNS name or the private IP address of this instance, update the information with the new configuration data.

**Note**

After you stop and restart the instance, the scheduled event is canceled immediately. The event's description is updated in the AWS console to reflect this.

### Instances Backed by Instance Store

If your instance's root device is an instance store volume and your instance is scheduled for network maintenance, power maintenance, stop, or retirement, you can opt to launch a replacement instance. After you have launched a replacement instance, you can terminate the original instance. If you choose to take this action, be aware that:

- The public DNS name and the private IP address of your replacement instance will differ from your original instance.
- You must first ensure that you have an AMI with the configuration that you want to use when launching the new instance. For more information, see:
  - [Creating an Instance Store-backed Windows AMI](#)
  - [Creating Instance Store-Backed Linux/UNIX AMIs \(p. 34\)](#)

## Ensuring Idempotency

When you launch an instance through the API or the command line interface tools (CLI), you can optionally provide a client token to ensure the request is idempotent. If timeouts or connection errors occur, you can repeat the request and be sure you haven't launched more instances than you intended. The token must be a unique, case-sensitive string of up to 64 ASCII characters.

If you repeat the request with the same client token, the same response is returned for each repeated request. The only information that might vary in the response is the state of the instance (for example, pending). The client token is included in the response when you describe the instance.

If you repeat the request with the same client token, but change another request parameter, Amazon EC2 returns an `IdempotentParameterMismatch` error.

The client token is valid for at least 24 hours after the termination of the instance. You should not reuse a client token for another call later on.

You can use the same client token for the same request across different regions. For example, if you send an idempotent request to launch an instance in the us-east-1 region, and you use the same exact call (with the same client token) in each of the other regions, the result is a different running instance in each region.

The following table shows common response codes and the recommended course of action.

Code	Retry	Comments
200 (OK)	No effect	After you receive a 200, the request has succeeded and any further retries have no effect.
400 (Client Error)	Not recommended	A 400 response typically indicates the request will never succeed (for example when a specified parameter value is not valid). In certain specific cases relating to resources that are transitioning between states, a repeat of the request could possibly succeed (e.g., launching an instance against an Amazon EBS volume that is currently transitioning to the available state).
500 (Server Internal Error)	Recommended	These errors are generally transient. Repeat the request with an appropriate back-off strategy.
503 (Server Unavailable)	Recommended	These errors can occur in times of extreme load. Repeat the request with an appropriate back-off strategy.

## Command Line Tools

Use the [ec2-run-instances](#) command.

### To send an idempotent request to launch instances

- Add the `--client-token` option to your request with a unique, case-sensitive string of up to 64 ASCII characters.

```
PROMPT> ec2-run-instances ami-b232d0db -k gsg-keypair --client-token
550e8400-e29b-41d4-a716-446655440000
```

## API

Use the [RunInstances](#) action.

### To send an idempotent request to launch instances

- Add the `ClientToken` parameter to your Query request with a unique, case-sensitive string of up to 64 ASCII characters.

```
https://ec2.amazonaws.com/
?Action=RunInstances
&ImageId=ami-3ac33653
...
&ClientToken=550e8400-e29b-41d4-a716-446655440000
&AUTHPARAMS
```

# Instance Metadata and User Data

## Topics

- [Retrieving Instance Metadata \(p. 366\)](#)
- [Retrieving User Data \(p. 368\)](#)
- [Example: AMI Launch Index Value \(p. 369\)](#)
- [Instance Metadata Categories \(p. 371\)](#)

*Instance metadata* is data about your EC2 instance that you can use to configure or manage the running instance. Instance metadata is divided into categories. For more information, see [Instance Metadata Categories \(p. 371\)](#).

EC2 instances can also include *dynamic data*, such as an instance identity document that is generated when the instance is launched. For more information, see [Dynamic Data Categories \(p. 375\)](#).

You can also access the *user data* that you supplied when launching your EC2 instance. For example, you can specify parameters for configuring your instance, or attach a simple script. You can also use this data to build more generic AMIs that can be modified by configuration files supplied at launch time. For example, if you run web servers for various small businesses, they can all use the same AMI and retrieve their content from the Amazon S3 bucket you specify in the user data at launch. To add a new customer at any time, simply create a bucket for the customer, add their content, and launch your AMI. If you launch more than one instance at the same time, the user data is available to all instances in that reservation.

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

## Important

Although you can only access instance metadata and user data from within the instance itself, the data is not protected by cryptographic methods. Anyone who can access the instance can view its metadata. Therefore, you should take suitable precautions to protect sensitive data (such as long-lived encryption keys). You should not store sensitive data, such as passwords, as user data.

For more information about adding user data when you launch an instance, see [Launching an Instance from an AMI \(p. 284\)](#). You can add or modify user data on Amazon EBS-backed instances when they're stopped. For more information about adding user data to a stopped instance, see [Modifying Attributes of a Stopped Instance \(p. 386\)](#).

When you are adding user data, take note of the following:

- User data is treated as opaque data: what you give is what you get back. It is up to the instance to be able to interpret it.
- User data is limited to 16 KB. This limit applies to the data in raw form, not base64-encoded form.
- User data must be base64-encoded before being submitted to the API. The API command line tools perform the base64 encoding for you. The data is decoded before being presented to the instance. For more information about base64 encodings, go to <http://tools.ietf.org/html/rfc4648>.

For more information about retrieving instance metadata and dynamic data, see [Retrieving Instance Metadata \(p. 366\)](#). For more information about retrieving user data, see [Retrieving User Data \(p. 368\)](#).

## Note

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

## Retrieving Instance Metadata

To view all categories of instance metadata from within a running instance, use the following URI:

```
http://169.254.169.254/latest/meta-data/
```

On a Linux instance, you can use a tool such as cURL, or use the GET command, for example:

```
PROMPT> GET http://169.254.169.254/latest/meta-data/
```

You can also download the Instance Metadata Query tool, which allows you to query the instance metadata without having to type out the full URI or category names:

<http://aws.amazon.com/code/1825>

On a Windows instance, you can install a tool such as GNU Wget or cURL to retrieve instance metadata at the command line, or you can copy and paste the URI into a browser. If you do not want to install any third-party tools, you can use PowerShell cmdlets to retrieve the URI. For example, if you are running version 3.0 or later of PowerShell, use the following cmdlet:

```
PROMPT> invoke-restmethod -uri http://169.254.169.254/latest/meta-data/
```

### Note

If you do install a third-party tool on a Windows instance, ensure that you read the accompanying documentation carefully, as the method of calling the HTTP and the output format might be different from what is documented here.

To retrieve dynamic data from within a running instance, use the following URI:

```
http://169.254.169.254/latest/dynamic/
```

All metadata is returned as text (content type text/plain). A request for a specific metadata resource returns the appropriate value, or a 404 – Not Found HTTP error code if the resource is not available.

A request for a general metadata resource (the URI ends with a /) returns a list of available resources, or a 404 – Not Found HTTP error code if there is no such resource. The list items are on separate lines, terminated by line feeds (ASCII 10).

## Examples of Retrieving Instance Metadata

The following are examples of requests and responses on a Linux instance.

This example gets the available versions of the instance metadata. These versions do not necessarily correlate with an Amazon EC2 API version. The earlier versions are available to you in case you have scripts that rely on the structure and information present in a previous version.

```
PROMPT> GET http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
```

```
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
latest
```

This example gets the top-level metadata items. Some items are only available for instances in a VPC. For more information about each of these items, see [Instance Metadata Categories \(p. 371\)](#).

```
PROMPT> GET http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
instance-action
instance-id
instance-type
kernel-id
local-hostname
local-ipv4
mac
network/
placement/
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
```

These examples get the value of some of the metadata items from the preceding example.

```
PROMPT> GET http://169.254.169.254/latest/meta-data/ami-id
ami-2bb65342
```

```
PROMPT> GET http://169.254.169.254/latest/meta-data/reservation-id
r-f ea54097
```

```
PROMPT> GET http://169.254.169.254/latest/meta-data/hostname
ec2-67-202-51-223.compute-1.amazonaws.com
```

This example gets the list of available public keys.

```
PROMPT> GET http://169.254.169.254/latest/meta-data/public-keys/
0=my-public-key
```

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

```
PROMPT> GET http://169.254.169.254/latest/meta-data/public-keys/0/
openssh-key
```

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

```
PROMPT> GET http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa MIICitCCAFICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRawDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC01BTsBDb25zb2x1MRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEg5vb25lQGFTYXpbvi5jb20wHhcNMTEwNDI1Mja0NTIxWhcN
MTIwNDI0Mja0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRawDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC01BTsBDb25z
b2x1MRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEg5vb25lQGFT
YXpbvi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMAk0dn+a4GmWIWJ
21uUSfwfEvySwtC2XADZ4nB+BLYgVlk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzzswY6786m86gpE
Ibb3OhjZnzcvQAArRhdlQWIMm2nrAgMBAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

This example shows the information available for a specific network interface (indicated by the MAC address) on an NAT instance in the EC2-Classic platform.

```
PROMPT> GET http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:29:96:8f:6a:2d/
device-number
local-hostname
local-ipv4s
mac
owner-id
public-hostname
public-ipv4s
```

This example gets the subnet ID for an EC2 instance launched into a VPC.

```
PROMPT> GET http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

## Retrieving User Data

To retrieve user data, use the following URI:

```
http://169.254.169.254/latest/user-data
```

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

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

```
PROMPT> GET http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

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

```
PROMPT> GET http://169.254.169.254/latest/user-data
[general]
instances: 4

[instance-0]
s3-bucket: <user_name>

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

## Example: AMI Launch Index Value

This example demonstrates how you can use both user data and instance metadata to configure your instances.

Alice wants to launch four instances of her favorite Linux database AMI, with the first acting as master and the remaining three acting as replicas. When she launches them, she wants to add user data about the replication strategy for each replicant. She is aware that this data will be available to all four instances, so she needs to structure the user data in a way that allows each instance to recognize which parts are applicable to it. She can do this using the `ami-launch-index` instance metadata value, which will be unique for each instance.

Here is the user data that Alice has constructed:

```
replicate-every=1min | replicate-every=5min | replicate-every=10min
```

The `replicate-every=1min` data defines the first replicant's configuration, `replicate-every=5min` defines the second replicant's configuration, and so on. Alice decides to provide this data as an ASCII string with a pipe symbol (|) delimiting the data for the separate instances.

Alice launches four instances, specifying the user data:

```
PROMPT> ec2-run-instances ami-2bb65342 -n 4 -d "replicate-every=1min | replicate-
every=5min | replicate-every=10min"

RESERVATION rfea54097 598916040194 default
INSTANCE i-10a64379 ami-2bb65342 pending 0 m1.small 2010-03-19T13:59:03+0000
us-east-1a aki-94c527fd ari-96c527ff monitoring-disabled ebs
INSTANCE i-10a64380 ami-2bb65342 pending 0 m1.small 2010-03-19T13:59:03+0000
us-east-1a aki-94c527fd ari-96c527ff monitoring-disabled ebs
INSTANCE i-10a64381 ami-2bb65342 pending 0 m1.small 2010-03-19T13:59:03+0000
us-east-1a aki-94c527fd ari-96c527ff monitoring-disabled ebs
INSTANCE i-10a64382 ami-2bb65342 pending 0 m1.small 2010-03-19T13:59:03+0000
us-east-1a aki-94c527fd ari-96c527ff monitoring-disabled ebs
```

After they're launched, all instances have a copy of the user data and the common metadata shown here:

- AMI id: ami-2bb65342
- Reservation ID: rfea54097
- Public keys: none
- Security group name: default
- Instance type: m1.small

However, each instance has certain unique metadata.

*Instance 1*

<b>Metadata</b>	<b>Value</b>
instance-id	i-10a64379
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*

<b>Metadata</b>	<b>Value</b>
instance-id	i-10a64380
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*

<b>Metadata</b>	<b>Value</b>
instance-id	i-10a64381
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*

<b>Metadata</b>	<b>Value</b>
instance-id	i-10a64382
ami-launch-index	3
public-hostname	ec2-67-202-51-226.compute-1.amazonaws.com
public-ipv4	67.202.51.226

Metadata	Value
local-hostname	ip-10-251-50-38.ec2.internal
local-ipv4	10.251.50.38

Alice can use the ami-launch-index value to determine which portion of the user data is applicable to a particular instance.

1. She connects to one of the instances, and retrieves the ami-launch-index for that instance to ensure it is one of the replicants:

```
PROMPT> GET http://169.254.169.254/latest/meta-data/ami-launch-index
2
```

2. She saves the ami-launch-index as a variable:

```
PROMPT> ami_launch_index=`GET http://169.254.169.254/latest/meta-data/ami-launch-index`
```

3. She saves the user data as a variable:

```
PROMPT> user_data=`GET http://169.254.169.254/latest/user-data/`
```

4. Finally, Alice runs a Linux cut command to extract the portion of the user data that is applicable to that instance:

```
PROMPT> echo $user_data | cut -d"|" -f"$ami_launch_index"
replicate-every=5min
```

## Instance Metadata Categories

The following table lists the categories of instance metadata.

Data	Description	Version Introduced
ami-id	The AMI ID used to launch the instance.	1.0
ami-launch-index	If you started more than one instance at the same time, this value indicates the order in which the instance was launched. The value of the first instance launched is 0.	1.0
ami-manifest-path	The path to the AMI's manifest file in Amazon S3. If you used an EBS-backed AMI to launch the instance, the returned result is unknown.	1.0

Data	Description	Version Introduced
ancestor-ami-ids	The AMI IDs of any instances that were rebundled to create this AMI. This value will only exist if the AMI manifest file contained an <code>ancestor-ami</code> key.	2007-10-10
block-device-mapping/ami	The virtual device that contains the root/boot file system.	2007-12-15
block-device-mapping/ebs <i>N</i>	The virtual devices associated with Amazon EBS volumes, if any are present. This value is only available in metadata if it is present at launch time. The <i>N</i> indicates the index of the Amazon EBS volume (such as <code>ebs1</code> or <code>ebs2</code> ).	2007-12-15
block-device-mapping/ephemeral <i>N</i>	The virtual devices associated with ephemeral devices, if any are present. The <i>N</i> indicates the index of the ephemeral volume.	2007-12-15
block-device-mapping/root	The virtual devices or partitions associated with the root devices, or partitions on the virtual device, where the root (/ or C:) file system is associated with the given instance.	2007-12-15
block-device-mapping/swap	The virtual devices associated with swap. Not always present.	2007-12-15
hostname	The private hostname of the instance. In cases where multiple network interfaces are present, this refers to the <code>eth0</code> device (the device for which the device number is 0).	1.0
iam/info	Returns information about the last time the instance profile was updated, including the instance's LastUpdated date, InstanceProfileArn, and InstanceProfileId.	2012-06-01
iam/security-credentials/ <i>role-name</i>	Where <i>role-name</i> is the name of the IAM role associated with the instance. Returns the temporary security credentials (AccessKeyId, SecretAccessKey, SessionToken, and Expiration) associated with the IAM role.	2012-06-01
instance-action	Notifies the instance that it should reboot in preparation for bundling. Valid values: <code>none</code>   <code>shutdown</code>   <code>bundle-pending</code> .	2008-09-01
instance-id	The ID of this instance.	1.0

Data	Description	Version Introduced
instance-type	The type of instance. For more information, see <a href="#">Instance Families and Types (p. 87)</a> .	2007-08-29
kernel-id	The ID of the kernel launched with this instance, if applicable.	2008-02-01
local-hostname	The private DNS hostname of the instance. In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0).	2007-01-19
local-ipv4	The private IP address of the instance. In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0).	1.0
mac	The instance's media access control (MAC) address. In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0).	2011-01-01
network/interfaces/macs/mac/device-number	The device number associated with that interface. Each interface must have a unique device number. The device number serves as a hint to device naming in the instance; for example, device-number is 2 for the eth2 device.	2011-01-01
network/interfaces/macs/mac/ipv4-associations/public-ip	The private IPv4 addresses that are associated with each public-ip address and assigned to that interface.	2011-01-01
network/interfaces/macs/mac/local-hostname	The interface's local hostname.	2011-01-01
network/interfaces/macs/mac/local-ipv4s	The private IP addresses associated with the interface.	2011-01-01
network/interfaces/macs/mac/mac	The instance's media access control (MAC) address.	2011-01-01
network/interfaces/macs/mac/owner-id	The ID of the owner of the network interface. In multiple-interface environments, an interface can be attached by a third party, such as Elastic Load Balancing. Traffic on an interface is always billed to the interface owner.	2011-01-01
network/interfaces/macs/mac/public-hostname	The interface's public hostname. Not returned for EC2 instances launched into a VPC.	2011-01-01

Data	Description	Version Introduced
network/interfaces/macs/mac/public-ipv4s	The elastic IP addresses associated with the interface. There may be multiple IP addresses on an instance.	2011-01-01
network/interfaces/macs/mac/security-groups	Security groups to which the network interface belongs. Returned only for EC2 instances launched into a VPC.	2011-01-01
network/interfaces/macs/mac/security-group-ids	IDs of the security groups to which the network interface belongs. Returned only for EC2 instances launched into a VPC. For more information on security groups in the EC2-VPC platform, see <a href="#">Security Groups for Your VPC</a> .	2011-01-01
network/interfaces/macs/mac/subnet-id	The ID of the subnet in which the interface resides. Returned only for EC2 instances launched into a VPC.	2011-01-01
network/interfaces/macs/mac/subnet-ipv4-cidr-block	The CIDR block of the subnet in which the interface resides. Returned only for EC2 instances launched into a VPC.	2011-01-01
network/interfaces/macs/mac/vpc-id	The ID of the VPC in which the interface resides. Returned only for EC2 instances launched into a VPC.	2011-01-01
network/interfaces/macs/mac/vpc-ipv4-cidr-block	The CIDR block of the VPC in which the interface resides. Returned only for EC2 instances launched into a VPC.	2011-01-01
placement/availability-zone	The Availability Zone in which the instance launched.	2008-02-01
product-codes	Product codes associated with the instance, if any.	2007-03-01
public-hostname	The public hostname of the instance. Not returned for EC2 instances launched into a VPC. For more information, see <a href="#">Elastic IP Addresses (EIP) (p. 438)</a> .	2007-01-19
public-ipv4	The public IP address. If an elastic IP address is associated with the instance, the value returned is the elastic IP address.	2007-01-19
public-keys/0/openssh-key	Public key. Only available if supplied at instance launch time.	1.0
ramdisk-id	The ID of the RAM disk specified at launch time, if applicable.	2007-10-10
reservation-id	ID of the reservation.	1.0

Data	Description	Version Introduced
security-groups	<p>The names of the security groups applied to the instance.</p> <p><b>Note</b>  Only EC2 instances launched into a VPC can change security groups after launch. These changes will be reflected here and in <code>networkInterfaces/macs/<i>mac</i>/securityGroups</code>.</p>	1.0

## Dynamic Data Categories

The following table lists the categories of dynamic data.

Data	Description	Version introduced
fws/instance-monitoring	Value showing whether the customer has enabled detailed one-minute monitoring in CloudWatch. Valid values: enabled   disabled	2009-04-04
instance-identity/document	JSON containing instance attributes, such as instance-id, private IP address, etc.	2009-04-04
instance-identity/pkcs7	Used to verify the document's authenticity and content against the signature.	2009-04-04
instance-identity/signature	Data that can be used by other parties to verify its origin and authenticity.	2009-04-04

# Stopping Instances

As soon as your instance starts to boot, you're billed for each hour or partial hour that you keep the instance running (even if the instance remains idle and you don't connect to it). You can stop instances that use Amazon EBS volumes as their root devices. When you stop an instance, it is shut down, so any data stored in the RAM of the host computer is not preserved. You are not billed for hourly usage or data transfer, but you are billed for the storage for any Amazon EBS volumes. You can restart the instance at any time with a start request. Each time you transition an instance from stopped to started, we charge a full instance hour, even if these transitions happen multiple times within a single hour.

When you've decided that you no longer need an instance, you can terminate it. As soon as the status of an instance changes to shutting down or terminated, you stop incurring charges for that instance.

For more information about pricing for Amazon EC2, see [Amazon EC2 Pricing](#).

### Topics

- [Instance Stop \(p. 376\)](#)
- [Instance Termination \(p. 377\)](#)
- [Stopping and Starting Instances \(p. 378\)](#)
- [Terminating Instances \(p. 380\)](#)
- [Enabling Termination Protection for an Instance \(p. 381\)](#)

- [Changing the Instance Initiated Shutdown Behavior \(p. 384\)](#)
- [Modifying Attributes of a Stopped Instance \(p. 386\)](#)

## Instance Stop

When you stop a running instance, the following happens:

- The instance performs a normal shutdown and stops running (the status goes to `stopping` and then `stopped`).
- The instance retains its instance ID.
- Any Amazon EBS volumes remain attached and the data persists.
- Any instance store volumes don't persist.
- The instance won't keep its public and private IP addresses. The exception is that instances launched in a VPC keep the same IP address when stopped and restarted.
- Any Elastic IP address mapped to the instance is unmapped. For instances launched into a VPC, the Elastic IP address remains associated with the stopped instance. When you restart the instance, the Elastic IP address is active. While the Elastic IP address is associated with the stopped instance, you're charged for the Elastic IP address.
- If you've registered the instance with a load balancer, it's likely that the load balancer won't be able to route traffic to your instance after you've stopped and restarted it. You must de-register the instance from the load balancer after stopping the instance, and then re-register after starting the instance. For more information, see [De-Registering and Registering Amazon EC2 Instances](#) section in the *Elastic Load Balancing Developer Guide*.
- When a Windows instance is stopped and restarted, the drive letters mapped to the attached EBS volumes are likely to change. For information on changing this default behavior, see [Using EC2Config](#) in the *Amazon Elastic Compute Cloud Microsoft Windows Guide*.

The next sections provide more detail about some of the areas covered in the preceding list.

### Network

When you stop an instance, we release its public and private IP addresses; we assign new IP addresses when you restart the instance. The exception is that instances launched into a VPC keep their private IP address when stopped and restarted. Therefore, it is possible to have a subnet with no running instances (they're all stopped), and also no remaining IP addresses available.

When you restart a Windows instance, by default, EC2ConfigService changes the instance host name to match the new IP address and initiates a reboot. For more information about this service, see [Using EC2Config](#) in the *Amazon Elastic Compute Cloud Microsoft Windows Guide*.

If you were using an Elastic IP address with the instance, the address is automatically unmapped when the instance stops. While the instance is stopped, you're charged for the address being unmapped (unless you remap it to another instance). When you restart the instance, you need to manually remap the Elastic IP address to the instance; it doesn't happen automatically.

For instances launched into a VPC, the Elastic IP address remains associated with the stopped instance. When you restart the instance, the Elastic IP address is active. While the Elastic IP address is associated with the stopped instance, you are charged for the Elastic IP address. For more information about the charges for Elastic IP addresses, see the [Amazon EC2 product page](#).

### Instance Attributes

Each Amazon EBS-backed instance has an attribute called `InstanceInitiatedShutdownBehavior` that controls whether the instance stops or terminates when you initiate a shutdown from within the

instance itself. The default is stop. You can modify the attribute while the instance is running or stopped. For more information, see [Changing the Instance Initiated Shutdown Behavior \(p. 384\)](#).

You can modify these instance attributes only while the instance is stopped:

- Kernel
- RAM disk
- Instance type
- User data

For more information about modifying instance attributes while the instance is stopped, see [Modifying Attributes of a Stopped Instance \(p. 386\)](#).

### **Storage**

While the instance is stopped, you can treat the root volume like any other volume, and modify it (for example, repair file system problems or update software). You just detach the volume from the stopped instance, attach it to a running instance, make your changes, detach it from the running instance, and then reattach it to the stopped instance. Make sure you reattach it to the correct storage device (whichever device name is specified as the root device in the instance's block device mapping).

While the instance is stopped, you can't change anything about the instance's block device mapping except the `DeleteOnTermination` flag for an attached volume (you can also change the flag while the instance is running). For more information, see [AWS Management Console \(p. 128\)](#).

## **Instance Termination**

When an instance terminates, any instance store associated with that instance is deleted.

By default, any volumes that you attach as you launch the instance are automatically deleted when the instance terminates. However, any volumes that you attached to a running instance persist even after the instance terminates. You can change the default behavior using the `DeleteOnTermination` flag. For an example of how to change this flag when launching an instance, see [Changing the Root Device Volume to Persist \(p. 128\)](#).

You can see the value for the `DeleteOnTermination` flag for each of the attached volumes by looking at the instance's block device mapping (for more information, see [Viewing the EBS Volumes in an Instance Block Device Mapping \(p. 530\)](#)). You can't change the flag's value after the instance is launched; however, you can launch the instance with your desired value for the flag. In the launch request, you just specify a block device mapping with the value. For an example of how to change the flag at launch time, see [AWS Management Console \(p. 128\)](#).

If you detach and then reattach a volume that was created at instance launch, it's treated like a new volume that you attached after the launch, and its `DeleteOnTermination` flag is set to `false`.

What happens to the volume when you terminate the instance it's attached to? Assuming the volume's `DeleteOnTermination` flag is `false`, the volume persists in its current state. You could take a snapshot of it, and you could attach it to any other instance you have.

### **Note**

When you launch an instance of a particular AMI and then terminate the instance, any remaining volumes are not automatically related to any future instances of that AMI. In other words, a new instance that you launch of that same AMI doesn't attempt to attach to the remaining volume. However, you can manually attach the remaining volume to the new instance if you want.

For more information on persisting the data after instance termination, see [Data Persistence after Instance Termination \(p. 459\)](#).

Starting with the 2009-10-31 API version, you can prevent an instance from being terminated. This feature is available for both Amazon EC2 instance store-backed and Amazon EBS-backed instances. Each instance has a `DisableApiTermination` attribute that defaults to `false` (the instance can be terminated). You can modify this instance attribute while the instance is running or stopped (in the case of Amazon EBS-backed instances).

## Stopping and Starting Instances

You can stop and restart Amazon EC2 instances that use Amazon EBS volumes as their root devices.

### Note

If you try to stop an instance that uses an instance store as its root device, you receive an error.

When an instance is stopped, it is shut down, and any data stored in RAM is not preserved. You're not billed for hourly usage or data transfer, but you're billed for any Amazon EBS volume storage. You can start the instance at any time with a start request. Each time you transition an instance from stopped to started, we charge a full instance hour, even if transitions happen multiple times within a single hour.

For information about the differences between stopping an instance and terminating an instance, see [Stopping Instances \(p. 375\)](#).

## AWS Management Console

### To stop and start an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Instances** in the **Navigation** pane.  
The console displays a list of running instances.
3. Select an instance, select **Instance Actions**, and click **Stop Instance**.  
A confirmation dialog box appears.
4. Click the **Yes, Stop Instance**.  
The instance is stopped and saved as a snapshot.
5. To restart the stopped instance, select **Stopped Instances** from the **Viewing** list box.  
The newly stopped instance appears in the list.
6. Select the instance, select **Instance Actions**, and click **Start Instance**.  
The instance begins restarting.

## Command Line Tools

### To stop and start an instance

1. Use the `ec2-stop-instances` command to stop an instance.

```
PROMPT> ec2-stop-instances i-10a64379
```

Amazon EC2 returns output similar to the following example.

```
IMAGE i-10a64379 running stopping
```

2. Use the `ec2-start-instances` command to restart the instance.

```
PROMPT> ec2-start-instances i-10a64379
```

Amazon EC2 returns output similar to the following example.

```
IMAGE i-10a64379 stopped pending
```

## API

### To stop and start an instance

1. Construct the following Query request to stop an instance.

```
https://ec2.amazonaws.com/
?Action=StopInstances
&InstanceId.1=i-10a64379
&AUTHPARAMS
```

The following is an example response.

```
<StopInstancesResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
 <requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId>
 <instancesSet>
 <item>
 <instanceId>exampleinstanceid</instanceId>
 <currentState>
 <code>64</code>
 <name>stopping</name>
 </currentState>
 <previousState>
 <code>16</code>
 <name>running</name>
 </previousState>
 </item>
 </instancesSet>
</StopInstancesResponse>
```

2. Construct the following Query request to restart the instance.

```
https://ec2.amazonaws.com/
?Action=StartInstances
&InstanceId.1=i-10a64379
&AUTHPARAMS
```

The following is an example response.

```
<StartInstancesResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
 <requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId>
 <instancesSet>
 <item>
 <instanceId>i-10a64379</instanceId>
 <currentState>
 <code>0</code>
```

```
<name>pending</name>
</currentState>
<previousState>
<code>80</code>
<name>stopped</name>
</previousState>
</item>
</instancesSet>
</StartInstancesResponse>
```

## Terminating Instances

This topic describes how to terminate an Amazon EC2 instance.

If the instance you launched is not in the free usage tier, as soon as your instance starts to boot, you're billed for each hour or partial hour that you keep the instance running (even if the instance is idle). When you've decided that you no longer need an instance, you can terminate it. As soon as the status of an instance changes to `shutting down` or `terminated`, you stop incurring charges for that instance.

For information about the differences between stopping an instance and terminating an instance, see [Stopping Instances \(p. 375\)](#). For information about preventing an instance from being terminated, see [Enabling Termination Protection for an Instance \(p. 381\)](#).

You cannot restart a terminated instance. However, you can launch additional instances of the same AMI.

## AWS Management Console

### To terminate an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Instances**.
3. Right-click the instance, and then click **Terminate**.
4. Click **Yes, Terminate** when prompted for confirmation.

## Command Line Tools

To terminate an instance, use the `ec2-terminate-instances` command. This example command terminates the instance with the ID `i-10a64379`.

```
ec2-terminate instances i-10a64379
```

The following is an example of the output for this command.

```
INSTANCE i-10a64379 running shutting-down
```

## API

To terminate an instance, use the [TerminateInstances](#) command.

## Enabling Termination Protection for an Instance

The following information applies to instances of Amazon EBS-backed AMIs or Amazon EC2 instance store-backed AMIs.

By default, any instances you launch can be externally terminated from the console, command line, or API. If you want to prevent accidental external termination of the instance, you can enable *termination protection* for the instance.

### Important

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

The instance's `DisableApiTermination` attribute controls whether the instance is protected from external termination. A value of `true` means the instance is protected and can't be externally terminated; a value of `false` means it's not protected and can be externally terminated. You can specify whether an instance is protected either at launch time or after the instance is running. You can modify this attribute while the instance is running or stopped (for Amazon EBS-backed instances).

### Note

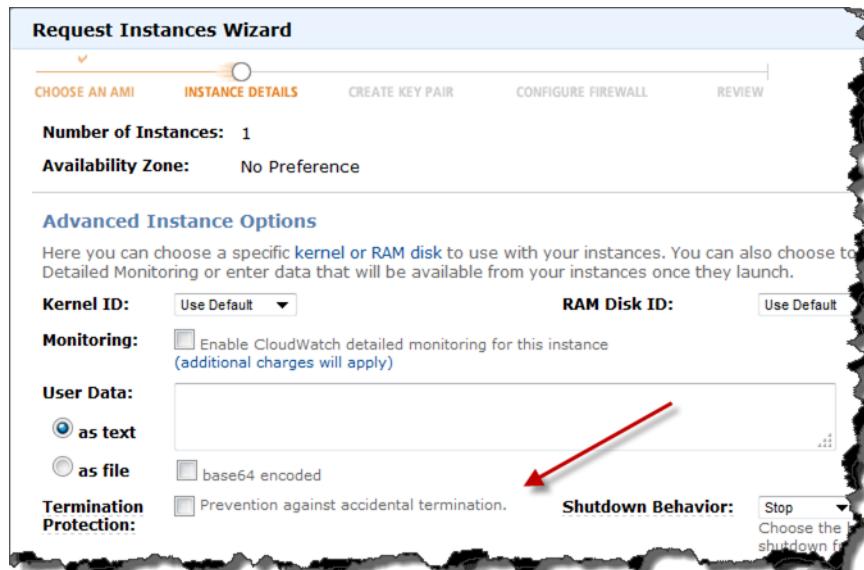
The `DisableApiTermination` attribute does not protect an instance from being terminated by being shutdown from within the instance when the `InstanceInitiatedShutdownBehavior` attribute is set for the instance. For more information about the `InstanceInitiatedShutdownBehavior` attribute, see [Changing the Instance Initiated Shutdown Behavior \(p. 384\)](#).

## AWS Management Console

By default, termination protection is disabled for an instance at launch time.

### To enable termination protection for an instance at launch time

- In the **Request Instances Wizard**, select the check box for **Termination Protection**.



### To enable termination protection for an instance after it's launched

1. In your list of instances, right-click the instance and select **Change Termination Protection**. A confirmation dialog box appears with the current status of termination protection.
2. Click **Yes, Enable**.

Now the instance cannot be terminated until you disable termination protection.

You can use the same procedure to disable termination protection for the instance. You can enable and disable this attribute of an instance as often as you want.

## Command Line Tools

By default, termination protection is disabled for an instance at launch time.

### To enable termination protection for an instance at launch time

- Add `--disable-api-termination` to the `ec2-run-instances` command.

```
PROMPT> ec2-run-instances ami_id --disable-api-termination ...
```

### To enable termination protection for an instance after it's launched

1. View the current value for the attribute with the following command.

```
PROMPT> ec2-describe-instance-attribute instance_id --disable-api-termination
```

The following is a sample response.

```
disableApiTermination i-87ad5eec false
```

2. Enable termination protection with the following command.

```
PROMPT> ec2-modify-instance-attribute instance_id --disable-api-termination true
```

The following is a sample response.

```
disableApiTermination i-87ad5eec true
```

### To disable termination protection for an instance

- Disable termination protection for the instance with the following command.

```
PROMPT> ec2-modify-instance-attribute instance_id --disable-api-termination false
```

The following is a sample response.

disableApiTermination	i-87ad5eec	false
-----------------------	------------	-------

You can enable and disable this attribute of an instance as often as you want.

## API

By default, termination protection is disabled for an instance at launch time.

### To enable termination protection for an instance at launch time

- Issue a Query request for RunInstances similar to the following example and include DisableApiTermination=true.

```
https://ec2.amazonaws.com/?Action=RunInstances
&ImageId=ami-60a54009
&MaxCount=3
&MinCount=1
&DisableApiTermination=true
&Placement.AvailabilityZone=us-east-1b
&Monitoring.Enabled=true
&AUTHPARAMS
```

For information about AUTHPARAMS, see [Common Query Parameters](#) in the *Amazon Elastic Compute Cloud API Reference*.

### To enable termination protection for an instance after it's launched

- Issue the following Query request to get the current value of the disableApiTermination attribute.

```
https://ec2.amazonaws.com/
?Action=DescribeInstanceAttribute
&InstanceId=i-87ad5eec
&Attribute=disableApiTermination
&AUTHPARAMS
```

The following is an example response.

```
<DescribeInstanceAttributeResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
 <requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId>
 <instanceId>i-87ad5eec</instanceId>
 <disableApiTermination>
 <value>false</value>
 </disableApiTermination>
</DescribeInstanceAttributeResponse>
```

- Issue the following Query request to modify the disableApiTermination attribute.

```
https://ec2.amazonaws.com/
?Action=ModifyInstanceAttribute
&InstanceId=i-87ad5eec
```

```
&DisableApiTermination.Value=true
&AUTHPARAMS
```

The following is an example response.

```
<ModifyInstanceAttributeResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/"> <requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId> <return>true</return></ModifyInstanceAttributeResponse>
```

### To disable termination protection for an instance

- Issue the following Query request to modify the `disableApiTermination` attribute.

```
https://ec2.amazonaws.com/
?Action=ModifyInstanceAttribute
&InstanceId=i-87ad5eec
&DisableApiTermination.Value=false
&AUTHPARAMS
```

The following is an example response.

```
<ModifyInstanceAttributeResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/"> <return>true</return></ModifyInstanceAttributeResponse>
```

You can enable and disable this attribute of an instance as often as you want.

## Changing the Instance Initiated Shutdown Behavior

By default, when you initiate a shutdown from within an EBS-backed instance, the instance stops. You can change this so the instance terminates instead. You just modify the `InstanceInitiatedShutdownBehavior` attribute for the instance. You can do this while the instance is either running or stopped.

### Note

This termination process will occur no matter what the `DisableApiTermination` attribute is set to. For more information about the `DisableApiTermination` attribute, see [Enabling Termination Protection for an Instance \(p. 381\)](#).

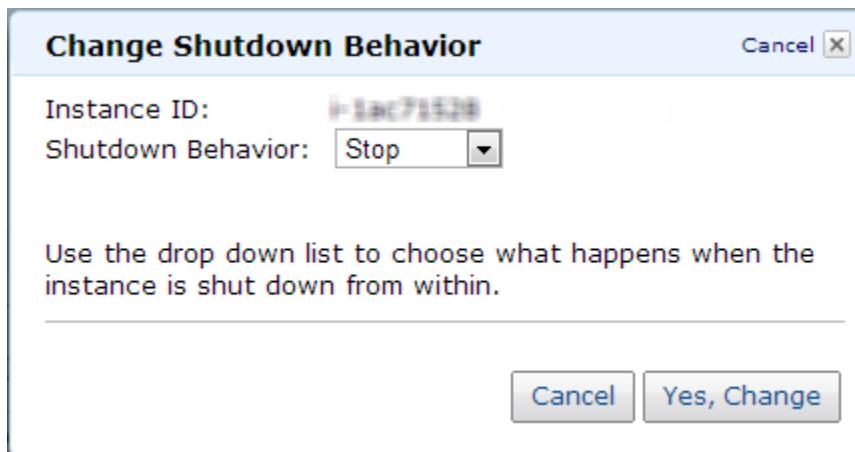
### Caution

When an AMI is created from an instance, the `InstanceInitiatedShutdownBehavior` attribute for the instance is also stored in the AMI. Any instances created from that AMI will inherit the `InstanceInitiatedShutdownBehavior` attribute. This can result in instances being inadvertently terminated when they are shut down from within the instance. You cannot change this attribute in the AMI, but you can change it in the launched instance. The best practice is to always set the `InstanceInitiatedShutdownBehavior` attribute of an instance to `stop` before creating an AMI from the instance.

## AWS Management Console

### To change the Shutdown Behavior

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Instances** in the navigation pane.
3. Context-click on the instance you want to modify and select **Change Shutdown Behavior**.
4. Select either **Terminate** or **Stop** from the **Shutdown Behavior** drop-down list.



5. Click **Yes, Change**.

You can toggle the attribute between **Stop** and **Terminate** as often as you want.

## Command Line tools

### To change the `instance-initiated-shutdown-behavior` attribute

1. View the current value for the attribute with the `ec2-describe-instance-attribute` command.

```
PROMPT> ec2-describe-instance-attribute instance_id --instance-initiated-shutdown-behavior
```

Sample response:

```
instanceInitiatedShutdownBehavior i-87ad5eec stop
```

2. Change the attribute value to terminate with the `ec2-modify-instance-attribute` command.

```
PROMPT> ec2-modify-instance-attribute instance_id --instance-initiated-shutdown-behavior terminate
```

Sample response:

```
instanceInitiatedShutdownBehavior i-87ad5eec terminate
```

You can toggle the attribute between `stop` and `terminate` as often as you want.

## API

### To change the `InstanceInitiatedShutdownBehavior` attribute

1. Call the [DescribeInstanceAttribute](#) action to get the current value of the attribute.

```
https://ec2.amazonaws.com/
?Action=DescribeInstanceAttribute
&InstanceId=i-87ad5eec
&Attribute=instanceInitiatedShutdownBehavior
&AUTHPARAMS
```

For information about AUTHPARAMS, see [Common Query Parameters](#) in the *Amazon Elastic Compute Cloud API Reference*.

The following is an example response.

```
<DescribeInstanceAttributeResponse xmlns="http://ec2.amazonaws.com/doc/2013-
02-01/">
 <requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId>
 <instanceId>i-5d7e8e36</instanceId>
 <instanceInitiatedShutdownBehavior>
 <value>stop</value>
 </instanceInitiatedShutdownBehavior>
</DescribeInstanceAttributeResponse>
```

2. Call the [ModifyInstanceAttribute](#) action to change the attribute's value.

```
https://ec2.amazonaws.com/
?Action=ModifyInstanceAttribute
&InstanceId=i-87ad5eec
&Attribute=instanceInitiatedShutdownBehavior
&Value=terminate
&AUTHPARAMS
```

The following is an example response.

```
<ModifyInstanceAttributeResponse xmlns="http://ec2.amazonaws.com/doc/2013-
02-01/">
 <requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId>
 <return>true</return>
</ModifyInstanceAttributeResponse>
```

You can toggle the attribute between `stop` and `terminate` as often as you want.

## Modifying Attributes of a Stopped Instance

Instances have a set of attributes, and you can modify the following attributes only when the instance is stopped:

- Kernel
- RAM disk
- Instance type
- User data

If you try to modify one of these attributes while the instance is running, Amazon EC2 returns the `IncorrectInstanceState` error.

For a list of all the available instance attributes you can change (not just those that require the instance to be stopped), see [ec2-modify-instance-attribute](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

**Note**

Although you can use the AWS Management Console to stop and start instances, and to change the instance type or user data, you can't use it to modify the kernel or RAM disk instance attributes. That functionality is available only through the command line tools or API.

For information about kernels and RAM disks, see [Kernel FAQ \(p. 84\)](#).

## AWS Management Console

### To change the instance type for a stopped instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Instances**.
3. In the **My Instances** pane, right-click a stopped instance, and then click **Change Instance Type**.
4. In the **Change Instance Type** dialog box, in the **Instance Type** drop-down list, select the type of instance you want, and then click **Yes, Change**.

### To change the user data for a stopped instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Instances**.
3. In the **My Instances** pane, right-click a stopped instance, and then click **View/Change User Data**.
4. In the **View/Change User Data** dialog box, update the user data, and then click **Yes, Change**.

## Command Line Tools

You can modify only a single attribute with each command.

### To change an attribute of a stopped instance

1. Choose one of the following commands to get the current value of the attribute of interest.

```
PROMPT> ec2-describe-instance-attribute instance_id --kernel
```

```
PROMPT> ec2-describe-instance-attribute instance_id --ramdisk
```

```
PROMPT> ec2-describe-instance-attribute instance_id --instance-type
```

```
PROMPT> ec2-describe-instance-attribute instance_id --user-data
```

The following is a sample response from the first command in the preceding list.

```
kernel i-87ad5eec aki-94c527fd
```

2. Choose one of the following commands to modify the value of the attribute of interest.

```
PROMPT> ec2-modify-instance-attribute instance_id --kernel kernel_id
```

```
PROMPT> ec2-modify-instance-attribute instance_id --ramdisk ramdisk_id
```

```
PROMPT> ec2-modify-instance-attribute instance_id --instance-type instance_type
```

```
PROMPT> ec2-modify-instance-attribute instance_id --user-data user_data
```

When you restart the instance, the new attribute value takes effect.

## API

You can modify only a single attribute in each call.

### To change an attribute of a stopped instance

1. Issue the following Query request to first get the current value of the attribute of interest. The example gets the current value for the instance type.

```
https://ec2.amazonaws.com/
?Action=DescribeInstanceAttribute
&InstanceId=i-87ad5eec
&Attribute=instanceType
&AUTHPARAMS
```

For information about AUTHPARAMS, see [Common Query Parameters](#) in the *Amazon Elastic Compute Cloud API Reference*.

The following is an example response.

```
<DescribeInstanceAttributeResponse xmlns="http://ec2.amazonaws.com/doc/2013-
02-01/">
 <requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId>
 <instanceId>i-5d7e8e36</instanceId>
 <instanceType>
 <value>m1.small</value>
 </instanceType>
</DescribeInstanceAttributeResponse>
```

2. Issue the following Query request to modify the attribute of interest. The example changes the instance type to c1.medium.

```
https://ec2.amazonaws.com/
?Action=ModifyInstanceAttribute
&InstanceId=i-87ad5eec
&Attribute=instanceType
&Value=c1.medium
&AUTHPARAMS
```

The following is an example response.

```
<ModifyInstanceAttributeResponse xmlns="http://ec2.amazonaws.com/doc/2013-
02-01/">>
 <return>true</return>
</ModifyInstanceAttributeResponse>
```

### Related Topics

- [Kernel FAQ \(p. 84\)](#)
- [Retrieving User Data \(p. 368\)](#)

## Troubleshooting Instances

### Topics

- [Troubleshooting Instances with Failed Status Checks \(p. 389\)](#)
- [What To Do If An Instance Immediately Terminates \(p. 412\)](#)
- [Getting Console Output and Rebooting Instances \(p. 413\)](#)
- [Instance Behavior When a Machine Fails \(p. 414\)](#)
- [Troubleshooting Instance Capacity \(p. 414\)](#)

This section provides information to help you troubleshoot problems with your instance.

## Troubleshooting Instances with Failed Status Checks

### Initial Steps You Can Take

#### Investigating Impaired Instances Using the AWS Management Console

If your instance fails a status check, first determine whether your applications are exhibiting any problems. If you verify that the instance is not running your applications as expected, follow these steps:

1. Log into the AWS Management Console.
2. In the **Amazon EC2** console, under My Instances, select your instance.
3. Click the **Status Checks** tab in the lower pane to see the individual results for all **System Status Checks** and **Instance Status Checks**.

If a system status check has failed, you can try one of the following options:

- For an instance using an EBS-backed AMI, stop and re-start the instance.
- For an instance using an instance-store backed AMI, terminate the instance and launch a replacement.
- Wait for our systems to resolve the issue.
- If these steps do not resolve your issue, you can seek assistance by posting your issue to the [Developer Forums](#) or contacting [Premium Support](#).

If an instance status check has failed, follow these steps:

1. Right-click your instance, and then click **Reboot**. It may take a few minutes for your system to restart.
2. Verify that the problem still exists; in some cases, rebooting may resolve the problem.
3. Wait until the instance shows a `running` state.
4. Right-click the instance and then click **Get System Log**. You can use this information to help identify the problem. Be sure you have rebooted recently to clear unnecessary information from the log.
5. Review the log that appears on the screen.
6. Use the list of known system log error statements below to troubleshoot your issue.
7. If these steps do not resolve your issue, you can seek assistance by posting your issue to the [Developer Forums](#) or contact [Premium Support](#).

If your experience differs from the our check results, or if you are having an issue with your instance that our checks did not detect, click **Submit Feedback** at the bottom of the **Status Checks** tab to help us improve our detection tests.

## Troubleshooting Instance Status Checks for Linux-Based Instances

For Linux-based instances that have failed an instance status check, such as the instance reachability check, first be sure you have followed the steps discussed earlier to retrieve the system log. The following list contains some common system log errors and suggested actions you can take to resolve the issue for each error .

### Memory Errors

- [Out of memory error \(OOM\): "Out of memory: kill process"](#) (p. 391)
- [Memory management update failed: "ERROR: mmu\\_update failed"](#) (p. 392)

### Device Errors

- ["I/O error, ... , sector " \(Block device failure\)](#) (p. 392)
- ["IO ERROR: neither local nor remote disk" \(Broken distributed block device\)](#) (p. 394)

### Kernel Errors

- ["request\\_module: runaway loop modprobe" \(Looping legacy kernel modprobe that applies to older Linux versions, e.g. 2.6.16-xenU\)](#) (p. 394)
- ["FATAL: kernel too old" and "fsck. ....: No such file or directory while trying to open /dev" \(Kernel and AMI mismatch\)](#) (p. 395)
- ["FATAL: Could not load /lib/modules" or "BusyBox" \(Missing kernel modules\)](#) (p. 396)
- ["ERROR Invalid kernel" \(EC2 incompatible kernel\)](#) (p. 397)

## File System Errors

- "request\_module: runaway loop modprobe" (Looping legacy kernel modprobe that applies to older Linux versions, e.g. 2.6.16-xenU) (p. 399)
- "fsck. ... : No such file or directory while trying to open /dev" (Filesystem not found) (p. 399)
- "General error mounting filesystems" (Failed Mount) (p. 401)
- "Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block" (Root filesystem mismatch) (p. 403)
- "Error: Unable to determine major/minor number of root device..." or "[ramfs /]#" (Root filesystem/device mismatch) (p. 404)
- "XENBUS: Device with no driver: device/vif0/"" (p. 405)
- Filesystem check required: "... days without being checked, check forced." (p. 406)
- "fsck died with exit status..." (Missing device). (p. 406)

## Operating System Errors

- GRUB prompt: "grubdom>" (p. 407)
- "Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. [FAILED]" (Hardcoded MAC address) (p. 410)
- "Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. ... Kernel panic - not syncing: Attempted to kill init!" (SELinux misconfiguration) (p. 411)
- "XENBUS: Timeout connecting to devices!" (Xenbus timeout) (p. 411)

## Memory Errors

### Out of memory error (OOM): "Out of memory: kill process"

An out of memory error is indicated by a system log entry similar to the one shown below:

```
[115879.769795] Out of memory: kill process 20273 (httpd) score 1285879
or a child
[115879.769795] Killed process 1917 (php-cgi) vsz:467184kB, anon-
rss:101196kB, file-rss:204kB
```

### Potential Cause

Exhausted memory

### Suggested Actions

For this instance type	Do this
EBS-backed	<p>Do one of the following:</p> <ul style="list-style-type: none"><li>• Stop the instance, and modify the instance to use a different instance type, and start the instance again. For example, you may want to use a larger instance type or a high-memory instance type.</li><li>• Reboot the instance to return it to a non-impaired status. The problem will probably occur again unless you change the instance type.</li></ul>

For this instance type	Do this
Instance-store backed	<p>Do one of the following:</p> <ul style="list-style-type: none"> <li>• Relaunch the instance using a different instance type. For example, you may want to use a larger instance type or a high-memory instance type.</li> <li>• Reboot the instance to return it to an unimpaired status. The problem will probably occur again unless you change the instance type.</li> </ul>

### Memory management update failed: "ERROR: mmu\_update failed"

Memory management update failures are indicated by a system log entry similar to the following:

```
...
Press `ESC' to enter the menu... 0 [H[J Booting 'Amazon Linux 2011.09
(2.6.35.14-95.38.amzn1.i686)'

root (hd0)

Filesystem type is ext2fs, using whole disk

kernel /boot/vmlinuz-2.6.35.14-95.38.amzn1.i686 root=LABEL=/ console=hvc0 LANG=
en_US.UTF-8 KEYTABLE=us

initrd /boot/initramfs-2.6.35.14-95.38.amzn1.i686.img

ERROR: mmu_update failed with rc=-22
```

### Potential Cause

Issue with Amazon Linux.

### Suggested Action

Seek assistance by posting your issue to the [Developer Forums](#) or contacting [Premium Support](#).

### Device Errors

#### "I/O error, ... , sector " (Block device failure)

An input/output error is indicated by a system log entry similar to the following example:

```
[9943662.053217] end_request: I/O error, dev sde, sector 52428288
[9943664.191262] end_request: I/O error, dev sde, sector 52428168
[9943664.191285] Buffer I/O error on device md0, logical block 209713024
[9943664.191297] Buffer I/O error on device md0, logical block 209713025
[9943664.191304] Buffer I/O error on device md0, logical block 209713026
[9943664.191310] Buffer I/O error on device md0, logical block 209713027
[9943664.191317] Buffer I/O error on device md0, logical block 209713028
```

```
[9943664.191324] Buffer I/O error on device md0, logical block 209713029
[9943664.191332] Buffer I/O error on device md0, logical block 209713030
[9943664.191339] Buffer I/O error on device md0, logical block 209713031
[9943664.191581] end_request: I/O error, dev sde, sector 52428280
[9943664.191590] Buffer I/O error on device md0, logical block 209713136
[9943664.191597] Buffer I/O error on device md0, logical block 209713137
[9943664.191767] end_request: I/O error, dev sde, sector 52428288
[9943664.191970] end_request: I/O error, dev sde, sector 52428288
[9943664.192143] end_request: I/O error, dev sde, sector 52428288
[9943664.192949] end_request: I/O error, dev sde, sector 52428288
[9943664.193112] end_request: I/O error, dev sde, sector 52428288
[9943664.193266] end_request: I/O error, dev sde, sector 52428288
...
...
```

## Potential Causes

Instance type	Potential cause
EBS-backed	Failed EBS volume
Instance-store backed	Failed physical drive

## Suggested Actions

For this instance type	Do this
EBS-backed	<p>Use one of the following procedures.</p> <ol style="list-style-type: none"><li>1. Stop the instance.</li><li>2. Detach the volume.</li><li>3. Attempt to recover the volume.</li><li>4. Re-attach the volume to the instance.</li><li>5. Detach the volume.</li></ol> <p>OR</p> <ol style="list-style-type: none"><li>1. Terminate the instance and relaunch the instance.</li></ol> <p><b>Note</b> Data will not be recovered.</p> <p><b>Tip</b> It's good practice to snapshot your EBS-backed volumes often. This dramatically decreases the risk of data loss as a result of failure.</p>

For this instance type	Do this
Instance-store backed	<p>Terminate the instance and relaunch the instance.</p> <p><b>Note</b> Data will not be recovered. Recover from backups</p> <p><b>Tip</b> It's a good practice to use either Amazon S3 or Amazon EBS volumes for backups. Non-EBS volumes are directly tied to single host and single disk failures.</p>

### "IO ERROR: neither local nor remote disk" (Broken distributed block device)

An input/output error on the device is indicated by a system log entry similar to the following example:

```
...
block drbd1: Local IO failed in request_timer_fn. Detaching...

Aborting journal on device drbd1-8.

block drbd1: IO ERROR: neither local nor remote disk

Buffer I/O error on device drbd1, logical block 557056

lost page write due to I/O error on drbd1

JBD2: I/O error detected when updating journal superblock for drbd1-8.
```

### Potential Causes

Instance type	Potential cause
EBS-backed	Failed EBS volume
Instance-store backed	Failed physical drive

### Suggested Action

Terminate the instance and launch another instance.

For an EBS-backed instance you can recover data from a recent snapshot by creating an image from it. Any data added after the snapshot will not be recovered.

### Kernel Errors

#### "request\_module: runaway loop modprobe" (Looping legacy kernel modprobe that applies to older Linux versions, e.g. 2.6.16-xenU)

This condition is indicated by a system log similar to the one shown below. Using an unstable or old Linux kernel can cause an interminable loop condition at startup.

```

Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007

BIOS-provided physical RAM map:

Xen: 0000000000000000 - 0000000026700000 (usable)

0MB HIGHMEM available.
...

request_module: runaway loop modprobe binfmt-464c

```

### Suggested Actions

For this instance type	Do this
EBS-backed	<p>Use a newer kernel, either Grub-based or static using one of the two procedures below.</p> <ul style="list-style-type: none"> <li>Relaunch the instance and specify the <code>-kernel</code> and <code>-ramdisk</code> parameters.</li> </ul> <p>OR</p> <ul style="list-style-type: none"> <li>Stop the instance.</li> <li>Modify the kernel and ramdisk attributes to use a newer kernel.</li> <li>Start the instance.</li> </ul>
Instance-store backed	Relaunch the instance and specify the <code>-kernel</code> and <code>-ramdisk</code> parameters.

### "FATAL: kernel too old" and "fsck. ....: No such file or directory while trying to open /dev" (Kernel and AMI mismatch)

This condition is indicated by a system log similar to the one shown below:

```

Linux version 2.6.16.33-xenU (root@dom0-0-50-45-1-a4-ee.z-2.aes0.internal)
(gcc version 4.1.1 20070105 (Red Hat 4.1.1-52)) #2 SMP Wed Aug 15 17:27:36 SAST
2007
...
FATAL: kernel too old
Kernel panic - not syncing: Attempted to kill init!

```

## Potential Causes

Incompatible kernel and userland.

## Suggested Actions

For this instance type	Do this
EBS-backed	<ol style="list-style-type: none"> <li>1. Modify the instance to use a newer kernel.             <ol style="list-style-type: none"> <li>a. Stop the instance.</li> <li>b. Modify the configuration to use a newer kernel.</li> <li>c. Start the instance.</li> </ol> </li> <li>2. Modify the AMI to use a newer kernel.</li> </ol>
Instance-store backed	<ol style="list-style-type: none"> <li>1. Modify the instance to use newer kernel.             <ol style="list-style-type: none"> <li>a. Stop the instance.</li> <li>b. Modify the configuration to use a newer kernel.</li> <li>c. Start the instance.</li> </ol> </li> <li>2. Modify the AMI to use a newer kernel.</li> </ol>

## "FATAL: Could not load /lib/modules" or "BusyBox" (Missing kernel modules)

This condition is indicated by a system log similar to the one shown below.

```
[0.370415] Freeing unused kernel memory: 1716k freed
Loading, please wait...
WARNING: Couldn't open directory /lib/modules/2.6.34-4-virtual: No such file
or directory
FATAL: Could not open /lib/modules/2.6.34-4-virtual/modules.dep.temp for writing:
No such file or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file
or directory
Couldn't get a file descriptor referring to the console
Begin: Loading essential drivers.... .
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file
or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file
or directory
Done.
Begin: Running /scripts/init-premount ...
Done.
Begin: Mounting root file system.... .
Begin: Running /scripts/local-top ...
Done.
Begin: Waiting for root file system.... .
Done.
Gave up waiting for root device. Common problems:
 - Boot args (cat /proc/cmdline)

```

```
- Check rootdelay= (did the system wait long enough?)
- Check root= (did the system wait for the right device?)
- Missing modules (cat /proc/modules; ls /dev)
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file
or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file
or directory
ALERT! /dev/sda1 does not exist. Dropping to a shell!
```

```
BusyBox v1.13.3 (Ubuntu 1:1.13.3-1ubuntu5) built-in shell (ash)
Enter 'help' for a list of built-in commands.

(initramfs)
```

## Potential Causes

One or more of the following conditions can cause this problem:

- Missing ramdisk
- Missing correct modules from ramdisk
- EBS root volume not correctly attached as /dev/sda1

## Suggested Actions

For this instance type	Do this
EBS-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none"><li>1. Select corrected ramdisk for the EBS volume.</li><li>2. Stop the instance.</li><li>3. Detach the volume and repair it.</li><li>4. Attach the volume to the instance.</li><li>5. Start the instance.</li><li>6. Modify the AMI to use the corrected ramdisk.</li></ol>
Instance-store backed	<p>Use the following procedure:</p> <ol style="list-style-type: none"><li>1. Terminate the instance and relaunch with correct ramdisk.</li><li>2. Rebundle the AMI with the corrected ramdisk definition.</li></ol>

## "ERROR Invalid kernel" (EC2 incompatible kernel)

This condition is indicated by a system log similar to the one shown below.

```
...
root (hd0)

Filesystem type is ext2fs, using whole disk

kernel /vmlinuz root=/dev/sdal ro

initrd /initrd.img

ERROR Invalid kernel: elf_xen_note_check: ERROR: Will only load images
built for the generic loader or Linux images
xc_dom_parse_image returned -1

Error 9: Unknown boot failure

Booting 'Fallback'

root (hd0)

Filesystem type is ext2fs, using whole disk

kernel /vmlinuz.old root=/dev/sdal ro

Error 15: File not found
```

## Potential Causes

One or both of the following conditions can cause this problem:

- Supplied kernel is not supported by Grub.
- Fallback kernel does not exist.

## Suggested Actions

For this instance type	Do this
EBS-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none"><li>1. Stop the instance.</li><li>2. Replace with working kernel.</li><li>3. Install a fallback kernel.</li><li>4. Modify the AMI by correcting the kernel.</li></ol>
Instance-store backed	<p>Use the following procedure:</p> <ol style="list-style-type: none"><li>1. Terminate and relaunch the instance with the correct kernel.</li><li>2. Rebundle AMI with the corrected kernel definition.</li><li>3. (Optional) Seek technical assistance for data recovery using <a href="#">Premium Support</a>.</li></ol>

## File System Errors

**"request\_module: runaway loop modprobe" (Looping legacy kernel modprobe that applies to older Linux versions, e.g. 2.6.16-xenU)**

This condition is indicated by a system log similar to the one shown below. Using an unstable or old Linux kernel can cause an interminable loop condition at startup.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007

BIOS-provided physical RAM map:

Xen: 0000000000000000 - 000000026700000 (usable)

0MB HIGHMEM available.
...
request_module: runaway loop modprobe binfmt-464c
```

## Suggested Actions

For this instance type	Do this
EBS-backed	<p>Use a newer kernel, either Grub-based or static using one of the two procedures below.</p> <ul style="list-style-type: none"><li>Relaunch the instance and specify the <code>-kernel</code> and <code>-ramdisk</code> parameters.</li></ul> <p>OR</p> <ul style="list-style-type: none"><li>Stop the instance.</li><li>Modify the kernel and ramdisk attributes to use a newer kernel.</li><li>Start the instance.</li></ul>
Instance-store backed	Relaunch the instance and specify the <code>-kernel</code> and <code>-ramdisk</code> parameters.

**"fsck. ... : No such file or directory while trying to open /dev" (Filesystem not found)**

This condition is indicated by a system log similar to the one shown below:

```
Welcome to Fedora
Press 'I' to enter interactive startup.
Setting clock : Wed Oct 26 05:52:05 EDT 2011 [OK]

Starting udev: [OK]

Setting hostname localhost: [OK]

No devices found
Setting up Logical Volume Management: File descriptor 7 left open
 No volume groups found
[OK]

Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1: clean, 82081/1310720 files, 2141116/2621440 blocks
[/sbin/fsck.ext3 (1) -- /mnt/dbbackups] fsck.ext3 -a /dev/sdh
fsck.ext3: No such file or directory while trying to open /dev/sdh

/dev/sdh:
The superblock could not be read or does not describe a correct ext2
filesystem. If the device is valid and it really contains an ext2
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
 e2fsck -b 8193 <device>

[FAILED]

*** An error occurred during the file system check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell.
Give root password for maintenance
(or type Control-D to continue):
```

## Potential Causes

- A bug exists in ramdisk filesystem definitions /etc/fstab
- Misconfigured filesystem definitions in /etc/fstab
- Missing/failed drive

## Suggested Actions

For this instance type	Do this
EBS-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none"> <li>1. Stop the instance, detach the root volume, repair/modify /etc/fstab the volume, attach the volume to the instance, and start the instance.</li> <li>2. Fix ramdisk to include modified /etc/fstab (if applicable).</li> <li>3. Modify the AMI to use a newer ramdisk.</li> </ol> <p><b>Tip</b>  The sixth field in the fstab defines availability requirements of the mount – a non-zero value implies that an fsck will be done on that volume and <i>must</i> succeed. Using this field can be problematic in Amazon EC2 because a failure typically results in an interactive console prompt which is not currently available in Amazon EC2. Use care with this feature and read the Linux man page for fstab.</p>
Instance-store backed	<p>Use the following procedure:</p> <ol style="list-style-type: none"> <li>1. Terminate the instance and relaunch another instance.</li> <li>2. Detach any errant EBS volumes and the reboot instance.</li> <li>3. (Optional) Seek technical assistance for data recovery using <a href="#">Premium Support</a>.</li> </ol>

## "General error mounting filesystems" (Failed Mount)

This condition is indicated by a system log similar to the one shown below.

```

Loading xenblk.ko module
xen-vbd: registered block device major 8

Loading ehci-hcd.ko module
Loading ohci-hcd.ko module
Loading uhci-hcd.ko module
USB Universal Host Controller Interface driver v3.0

Loading mbcache.ko module
Loading jbd.ko module
Loading ext3.ko module
Creating root device.
Mounting root filesystem.
kjournald starting. Commit interval 5 seconds

```

```
EXT3-fs: mounted filesystem with ordered data mode.

Setting up other filesystems.
Setting up new root fs
no fstab.sys, mounting internal defaults
Switching to new root and running init.
unmounting old /dev
unmounting old /proc
unmounting old /sys
mountall:/proc: unable to mount: Device or resource busy
mountall:/proc/self/mountinfo: No such file or directory
mountall: root filesystem isn't mounted
init: mountall main process (221) terminated with status 1

General error mounting filesystems.
A maintenance shell will now be started.
CONTROL-D will terminate this shell and re-try.
Press enter for maintenance
(or type Control-D to continue):
```

## Potential Causes

Instance type	Potential cause
EBS-backed	<ul style="list-style-type: none"><li>Detached or failed EBS volume.</li><li>Corrupted filesystem.</li><li>Mismatched ramdisk and AMI combination (e.g., Debian ramdisk with a Suse AMI).</li></ul>
Instance-store backed	<ul style="list-style-type: none"><li>Failed drive.</li><li>Corrupted file system.</li><li>Mismatched ramdisk and combination (for example, Debian ramdisk with a Suse AMI).</li></ul>

## Suggested Actions

For this instance type	Do this
EBS-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none"> <li>1. Stop the instance.</li> <li>2. Detach the root volume.</li> <li>3. Attach the root volume to a known working instance.</li> <li>4. Run filesystem check (<code>fsck -a /dev/...</code>).</li> <li>5. Fix any errors.</li> <li>6. Detach the volume from the known working instance.</li> <li>7. Attach the volume to the stopped instance.</li> <li>8. Start the instance.</li> <li>9. Recheck the instance status.</li> </ol>
Instance-store backed	<p>Try one of the following:</p> <ul style="list-style-type: none"> <li>• Restart a new instance.</li> <li>• (Optional) Seek technical assistance for data recovery using <a href="#">Premium Support</a>.</li> </ul>

### "Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block" (Root filesystem mismatch)

This condition is indicated by a system log similar to the one shown below.

```

Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
Kernel command line: root=/dev/sdal ro 4
...
Registering block device major 8
...
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(8,1)

```

## Potential Causes

Instance type	Potential cause
EBS-backed	<ul style="list-style-type: none"> <li>• Device not attached correctly.</li> <li>• Root device not attached at correct device point.</li> <li>• Filesystem not expected format.</li> <li>• Use of legacy kernel (e.g., 2.6.16-XenU).</li> </ul>

Instance type	Potential cause
Instance-store backed	Hardware device failure.

### Suggested Actions

For this instance type	Do this
EBS-backed	Do one of the following: <ul style="list-style-type: none"> <li>• Stop and then restart the instance.</li> <li>• Modify root volume to attach at the correct device point, possibly /dev/sda1 instead of /dev/sda.</li> <li>• Stop and modify to use modern kernel.</li> </ul>
Instance-store backed	Terminate and relaunch the instance. We strongly recommend using a modern kernel.

### "Error: Unable to determine major/minor number of root device..." or "[ramfs /]#" (Root filesystem/device mismatch)

This condition is indicated by a system log similar to the one shown below.

```

...
XENBUS: Device with no driver: device/vif/0
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udevd[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
 Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#

```

### Potential Causes

- Missing or incorrectly configured virtual block device driver
- Device enumeration clash (sda versus xvda or sda instead of sda1)
- Incorrect choice of DomU kernel

## Suggested Actions

For this instance type	Do this
EBS-backed	Use the following procedure: <ol style="list-style-type: none"> <li>1. Stop the instance.</li> <li>2. Detach the volume.</li> <li>3. Fix the device mapping problem.</li> <li>4. Start the instance.</li> <li>5. Modify the AMI to address device mapping issues.</li> </ol>
Instance-store backed	Use the following procedure: <ol style="list-style-type: none"> <li>1. Rebundle with the appropriate fix (map block device correctly).</li> <li>2. Terminate and relaunch the instance.</li> </ol>

## "XENBUS: Device with no driver: device/vif/0"

This condition is indicated by a system log similar to the one shown below.

```

XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udevd[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
 Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#

```

## Potential Causes

- Missing or incorrectly configured virtual block device driver
- Device enumeration clash (sda versus xvda)
- Incorrect choice of DomU kernel

## Suggested Actions

For this instance type	Do this
EBS-backed	Use the following procedure: <ol style="list-style-type: none"> <li>1. Stop the instance.</li> <li>2. Detach the volume.</li> <li>3. Fix the device mapping problem.</li> <li>4. Start the instance.</li> <li>5. Modify the AMI to address device mapping issues.</li> </ol>
Instance-store backed	Use the following procedure: <ol style="list-style-type: none"> <li>1. Rebundle with the appropriate fix (map block device correctly).</li> <li>2. Terminate and relaunch the instance.</li> </ol>

### Filesystem check required: "... days without being checked, check forced."

This condition is indicated by a system log similar to the one shown below.

```
...
Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1 has gone 361 days without being checked, check forced
```

## Potential Causes

Filesystem check time passed; a filesystem check is being forced.

## Suggested Actions

- Wait until the filesystem check completes. Note that a filesystem check can take a long time depending on the size of the root filesystem.
- Modify your filesystems to remove the filesystem check (fsck) enforcement using tune2fs or tools appropriate for your filesystem.

### "fsck died with exit status..." (Missing device).

This condition is indicated by a system log similar to the one shown below.

```
Cleaning up ifupdown....
Loading kernel modules...done.
...
Activating lvm and md swap...done.
Checking file systems...fsck from util-linux-ng 2.16.2
```

```
/sbin/fsck.xfs: /dev/sdh does not exist
fsck died with exit status 8
[31mfailed (code 8).[39;49m
```

## Potential Causes

- Ramdisk looking for missing drive
- Filesystem consistency check forced
- Drive failed or detached

## Suggested Actions

For this instance type	Do this
EBS-backed	<p>Try one or more of the following to resolve the issue:</p> <ul style="list-style-type: none"><li>• Stop the instance, attach volume to existing running instance.</li><li>• Manually run consistency checks.</li><li>• Fix ramdisk to include relevant utilities.</li><li>• Modify filesystem tuning parameters to remove consistency requirements (not recommended).</li></ul>
Instance-store backed	<p>Try one or more of the following to resolve the issue:</p> <ul style="list-style-type: none"><li>• Rebundle ramdisk with correct tooling.</li><li>• Modify filesystem tuning parameters to remove consistency requirements (not recommended).</li><li>• Terminate and restart.</li><li>• (Optional) Seek technical assistance for data recovery using <a href="#">Premium Support</a>.</li></ul>

## Operating System Errors

### GRUB prompt: "grubdom>"

This condition is indicated by a system log similar to the one shown below.

```
GNU GRUB version 0.97 (629760K lower / 0K upper memory)

[Minimal BASH-like line editing is supported. For
the first word, TAB lists possible command
completions. Anywhere else TAB lists the possible
```

```
completions of a device/filename.]
```

*grubdom>*

## Potential Causes

Instance type	Potential causes
EBS-backed	<ul style="list-style-type: none"><li>• Missing grub.conf file.</li><li>• Incorrect Grub image used expecting grub.conf at different location.</li><li>• Unsupported filesystem used to store grub.conf.</li></ul>
Instance-store backed	<ul style="list-style-type: none"><li>• Missing grub.conf file.</li><li>• Incorrect Grub image used expecting grub.conf at different location.</li></ul>

### Suggested Actions

For this instance type	Do this
EBS-backed	<p>Use one of the following procedures:</p> <p>Modify the AMI and relaunch the instance</p> <ol style="list-style-type: none"> <li>1. Modify the source AMI to create a grub.conf at the standard location (/boot/grub/menu.lst).</li> <li>2. Pick the appropriate Grub image, (hd0-1st drive or hd00 – 1st drive, 1st partition).</li> <li>3. Relaunch the instance.</li> </ol> <p>OR</p> <p>Fix the existing instance</p> <ol style="list-style-type: none"> <li>1. Stop the instance.</li> <li>2. Detach the root filesystem.</li> <li>3. Attach the root filesystem to a known working instance.</li> <li>4. Mount filesystem.</li> <li>5. Create grub.conf.</li> <li>6. Detach filesystem.</li> <li>7. Attach to the original instance.</li> <li>8. Modify kernel attribute to use appropriate Grub (1st disk or 1st partition on 1st disk).</li> <li>9. Start the instance.</li> </ol>
Instance-store backed	<p><b>Option 1: Modify the AMI and relaunch the instance</b></p> <ol style="list-style-type: none"> <li>1. Modify the source AMI to create a grub.conf at the standard location (/boot/grub/menu.lst).</li> <li>2. Pick the appropriate Grub image, (hd0-1st drive or hd00 – 1st drive, 1st partition).</li> <li>3. Relaunch the instance.</li> </ol> <p><b>Option 2: Fix the existing instance</b></p> <p>Relaunch the instance using the correct kernel if you believe the kernel was incorrect.</p> <p><b>Note</b>  To recover data from the existing instance, contact AWS Premium Support.</p>

## "Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. [FAILED]" (Hardcoded MAC address)

This condition is indicated by a system log similar to the one shown below:

```
...
Bringing up loopback interface: [OK]
Bringing up interface eth0: Device eth0 has different MAC address than expected,
ignoring.
[FAILED]
Starting auditd: [OK]
```

### Potential Causes

There is a hardcoded interface MAC in the AMI configuration.

### Suggested Actions

For this instance type	Do this
EBS-backed	<p>Do one of the following:</p> <ul style="list-style-type: none"><li>• Modify the AMI to remove the hard coding and relaunch the instance.</li></ul> <p>OR</p> <ul style="list-style-type: none"><li>• Modify the instance to remove the hard coded MAC address.</li></ul> <p>OR</p> <ol style="list-style-type: none"><li>1. Stop the instance.</li><li>2. Detach the root file system.</li><li>3. Modify the instance to remove the hard coded MAC address.</li><li>4. Attach the volume to original instance.</li><li>5. Start the instance.</li></ol>
Instance-store backed	<p>Do one of the following:</p> <ul style="list-style-type: none"><li>• Modify the instance to remove the hard coded MAC address.</li><li>• Relaunch the instance.</li></ul>

## "Unable to load SELinux Policy. Machine is in enforcing mode. Halting now... Kernel panic - not syncing: Attempted to kill init!" (SELinux misconfiguration)

This condition is indicated by a system log similar to the one shown below.

```
audit(1313445102.626:2): enforcing=1 old_enforcing=0 auid=4294967295
Unable to load SELinux Policy. Machine is in enforcing mode. Halting now.
Kernel panic - not syncing: Attempted to kill init!
```

### Potential Causes

SELinux has been enabled in error:

- Supplied kernel is not supported by Grub.
- Fallback kernel does not exist.

### Suggested Actions

For this instance type	Do this
EBS-backed	Use the following procedure: <ol style="list-style-type: none"><li>1. Stop the instance.</li><li>2. Detach the volume.</li><li>3. Disable SELinux.</li><li>4. Start the instance.</li></ol>
Instance-store backed	Use the following procedure: <ol style="list-style-type: none"><li>1. Terminate and relaunch the instance.</li><li>2. (Optional) Seek technical assistance for data recovery using <a href="#">Premium Support</a>.</li></ol>

## " XENBUS:Timeout connecting to devices!" ((Xenbus timeout))

This condition is indicated by a system log similar to the one shown below.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
XENBUS: Timeout connecting to devices!
...
Kernel panic - not syncing: No init found. Try passing init= option to kernel.
```

### Potential Causes

- The block device not is connected to the instance.
- This instance is using a very old DomU kernel.

### Suggested Actions

For this instance type	Do this
EBS-backed	Do one of the following: <ul style="list-style-type: none"><li>• Modify AMI and instance to use a modern kernel and relaunch the instance.</li><li>• Reboot the instance.</li></ul>
Instance-store backed	Do one of the following: <ul style="list-style-type: none"><li>• Reboot the instance.</li><li>• Modify AMI and instance to use a modern kernel and relaunch the instance.</li></ul>

## What To Do If An Instance Immediately Terminates

We recommend that when you launch an instance, you immediately check its status to confirm that it goes to the running status and not terminated.

The following are a few reasons why an Amazon EBS-backed instance might immediately terminate:

- You've reached your volume limit. For information about the volume limit, and to submit a request to increase your volume limit, see [Request to Increase the Amazon EBS Volume Limit](#).
- The AMI is missing a required part.
- The snapshot is corrupt.

This section describes how to get information about the reason that the instance terminated.

### AWS Management Console

#### To get information about the reason that the instance terminated

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the **Navigation** pane, click **Instances** to display the instance details.
3. Select your instance.
4. In the **Description** tab, locate the reason next to the label **State Transition Reason**. If the instance has remained running since it was launched, typically no reason is listed. If you've explicitly stopped or terminated the instance, the reason is `User initiated shutdown`.

### Command Line Tools

#### To get information about why the instance terminated

1. Use the **ec2-describe-instances** command in verbose mode:

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

2. In the XML response that's displayed, locate the `stateReason` element. It looks similar to the following example.

```
<stateReason>
 <code>Client.UserInitiatedShutdown</code>
 <message>Client.UserInitiatedShutdown: User initiated shutdown</message>
</stateReason>
```

The preceding example shows what's displayed for a normal instance that you've stopped. For the instance that terminated immediately, the `code` and `message` elements will describe the reason for the termination (e.g., `VolumeLimitExceeded`).

## API

### To get information about why the instance terminated

1. Construct the following Query request. For information about AUTHPARAMS, see [Common Query Parameters](#) in the *Amazon Elastic Compute Cloud API Reference*.

```
https://ec2.amazonaws.com/
?Action=DescribeInstances
&AUTHPARAMS
```

2. In the XML response that's displayed, locate the `stateReason` element. It looks similar to the following example.

```
<stateReason>
 <code>Client.UserInitiatedShutdown</code>
 <message>Client.UserInitiatedShutdown: User initiated shutdown</message>
</stateReason>
```

The preceding example shows what's displayed for a normal instance that you've stopped. For the instance that terminated immediately, the `code` and `message` elements will describe the reason for the termination (e.g., `VolumeLimitExceeded`).

## 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 API and the command line interface tools (CLI).

## Console Output

For Linux/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 instance's owner can retrieve it.

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

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

**Note**

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

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

**Note**

Only the instance owner can access the console output.

## Instance Reboot

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

**Caution**

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

## Instance Behavior When a Machine Fails

If there are instances running on a machine which fails, the instances terminate and need to be relaunched. All data on the instance stores and on Amazon EBS root volumes (for Amazon EBS-backed instances) is also lost by default.

The default behavior of the root volume on the Amazon EBS-backed instance can be changed by setting the value of `deleteOnTermination` to `false`. For more information, see [Amazon EC2 Root Device Volume \(p. 123\)](#). The data on the non-root Amazon EBS volumes (if attached) is preserved, by default, on instance termination.

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

## Troubleshooting Instance Capacity

If you get an "InsufficientInstanceCapacity" error when you try to launch an instance, AWS does not currently have enough available capacity to service your request. 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.

If you get an "InstanceLimitExceeded" error when you try to launch an instance, you have reached your concurrent running instance limit. For new AWS accounts, the limit is 20. If you need additional capacity, please complete the form at [Request to Increase Amazon EC2 Instance Limit](#).

# Network and Security

---

## Topics

- [Amazon EC2 Security Groups \(p. 415\)](#)
- [Controlling Access to Amazon EC2 Resources \(p. 421\)](#)
- [Amazon EC2 and Amazon Virtual Private Cloud \(VPC\) \(p. 428\)](#)
- [Amazon EC2 Instance IP Addressing \(p. 430\)](#)
- [Elastic IP Addresses \(EIP\) \(p. 438\)](#)
- [Elastic Network Interfaces \(ENI\) \(p. 442\)](#)

This section describes key network and security features related to Amazon EC2.

## Amazon EC2 Security Groups

A *security group* acts as a firewall that controls the traffic allowed to reach one or more instances. When you launch an instance, you assign it one or more security groups. You add rules to each security group that control traffic for the instance. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances to which the security group is assigned.

## Topics

- [Security Group Concepts \(p. 415\)](#)
- [Creating a Security Group \(p. 419\)](#)
- [Describing Security Groups \(p. 419\)](#)
- [Adding a Security Group Rule \(p. 419\)](#)
- [Deleting a Security Group Rule \(p. 420\)](#)
- [Deleting a Security Group \(p. 420\)](#)
- [API and Command Overview \(p. 421\)](#)

## Security Group Concepts

This section provides information about security groups and security group rules.

## Topics

- Security Groups for EC2-Classic (p. 416)
- Security Groups for EC2-VPC (p. 416)
- Default Security Group (p. 416)
- Creating Your Own Security Groups (p. 417)
- Instance Group Membership (p. 417)
- Security Group Rules (p. 417)

## Security Groups for EC2-Classic

If you're using EC2-Classic, you must create security groups for each region in which you'll launch instances. When you launch an instance in EC2-Classic, you must specify a security group in the same region as the instance. You can't specify any security groups you created for EC2-VPC when you launch an instance in EC2-Classic.

Security group rules control the inbound traffic allowed to reach the instances associated with the security group. All other inbound traffic is discarded, and all outbound traffic is allowed by default.

**Note**

On EC2-Classic, you can't change which security groups are assigned to an instance after it's launched. However, on EC2-VPC, you can change which security groups are assigned to an instance after it's launched.

## Security Groups for EC2-VPC

If you're using EC2-VPC, you must create security groups specifically for your VPC. You can only specify the security groups for a VPC with instances in that VPC. You can't specify any security groups you created for EC2-Classic when you launch an instance in a VPC.

Security group rules control the inbound traffic allowed to reach the instances associated with the security group and the outbound traffic that can reach the Internet. All other traffic is discarded.

When you specify a security group for a nondefault VPC to the CLI or the API actions, you must use the security group ID and not the security group name to identify the security group.

Security groups for EC2-VPC have additional capabilities that aren't supported by security groups for EC2-Classic. For more information about security groups for EC2-VPC, see [Security Groups for Your VPC in the Amazon Virtual Private Cloud User Guide](#).

## Default Security Group

Your AWS account automatically comes with a *default security group* for EC2-Classic. When you create a VPC, it automatically comes with a default security group. If you don't specify a different security group when you launch an instance, the instance is automatically associated with the appropriate default security group.

A default security group is named `default`, and it has an ID assigned by AWS. These are the initial settings for the default security group:

- Allow no inbound traffic (other than from instances with the same security group)
- Allow all outbound traffic from the instance
- Allow all inbound and outbound traffic between instances with the same security group

You can change the rules for a default security group. For example, you can add an inbound rule to allow SSH or Remote Desktop connections from specific hosts for the purposes of instance management.

You can't delete a default security group.

## Creating Your Own Security Groups

If you don't want all your instances to use the default security group, you can create your own security groups. You can create security groups that reflect the different roles your instances play in your system. For information about security groups for web servers and database servers, see [Creating the WebServerSG and DBServerSG Security Groups](#) in the *Amazon Virtual Private Cloud User Guide*.

When you create a security group, you must provide it with a name and a description. Security group names and descriptions can be up to 255 characters in length, and are limited to the following characters:

- EC2-Classic: ASCII characters
- EC2-VPC: a-z, A-Z, 0-9, spaces, and .\_-:/()#,@[]+=&:{}!\$\*

AWS assigns each security group a unique ID in the form sg-xxxxxxx. The following are the initial settings for a security group that you create:

- Allow no inbound traffic
- Allow all outbound traffic

After you've created a security group, you can change its inbound rules to reflect the type of inbound traffic that you want to reach the associated instances. In EC2-VPC, you can also change its outbound rules. To allow instances that have the same security group to talk to each other, you must explicitly add rules for this.

**Note**

In EC2-Classic, you can create up to 500 security groups in each region for each account. In EC2-VPC, you can create up to 100 security groups per VPC. The security groups for EC2-Classic do not count against the security group limit for EC2-VPC.

## Instance Group Membership

When you launch an instance, you can associate it with one or more security groups. When deciding whether to allow traffic to reach an instance, we evaluate all the rules from all the security groups associated with the instance.

**Note**

In EC2-Classic, you can assign up to 500 security groups to an instance. In EC2-VPC, you can assign up to 5 security groups to an instance.

If you don't specify any security groups when you launch an instance, the instance is automatically associated with the default security group and uses the rules for that group.

After you launch an instance in EC2-Classic, you can't change its security groups. However, you can change the rules of a security group, and those changes are automatically applied to all instances that are associated with the security group.

## Security Group Rules

The rules of a security group control the inbound traffic that's allowed to reach the instances associated with the security group. All outbound traffic is allowed by default. You can't change the outbound behavior for EC2-Classic.

You can't modify an existing rule in a security group. However, you can add and remove rules at any time. Your changes are automatically applied to instances associated with the security group after a short period.

For each rule, you define two settings:

- The protocol allowed.
- The source that defines the access to the port. The source can be one of the following:
  - An individual IP address (for example, 203.0.113.1).
  - An IP address range, in CIDR notation (for example, 203.0.113.0/24).
  - A security group. This security group can be one of the following:
    - The current security group.
    - A different security group in your AWS account.
    - A security group in another AWS account.

Each inbound security group rule enables a specific source to reach the instances using a certain protocol (TCP, UDP, or ICMP) and destination port or ports (if the protocol is TCP or UDP). For example, a rule could allow IP address 203.0.113.1 (the source) to reach the instances on TCP port 22 (the protocol and destination port). If you specify ICMP as the protocol for the rule, you must also specify an ICMP type and code.

**Note**

You can get the public IP address of your local computer using a service. To locate a service that provides your IP address, use the search phrase "what is my IP address". If you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

By specifying a security group as the source, you allow incoming traffic to reach all instances associated with the source security group. The incoming traffic that you allow is based on the private IP addresses of the instances associated with the source security group. The security group can be another security group in your AWS account, a security group in a different AWS account, or the security group itself. If the security group is in a different AWS account, prefix the security group name with the AWS account ID and a forward slash (for example, 111122223333/sg-edcd9784).

**Note**

You can't create a security group for EC2-VPC that references a security group for EC2-Classic, and you can't create a security group for EC2-Classic that references a security group for EC2-VPC.

If you specify the security group itself as the source, each instance accepts inbound traffic from other instances associated with the security group. For example, the default security group specifies itself as a source security group in its inbound security group rules. This is why instances associated with the default security group allow inbound traffic from other instances associated with the default security group.

Security group rules are always permissive, so you cannot create any rules that deny access to a port. Also, if there is more than one rule for a specific port, the most permissive rule is always applied. For example, if you have a rule that allows access to port 22 from IP address 203.0.113.1, and another rule that allows access to port 22 from everyone, everyone has access to port 22.

When multiple security groups are applied to an instance, the rules from each security group are effectively aggregated to create one larger set of rules. The aggregated rules are then used to determine access to the instances.

In EC2-Classic, you can add up to 100 rules per security group. In EC2-VPC, you can add up to 50 rules per security group. If you need additional rules for an instance, you can assign it multiple security groups.

**Caution**

Because you can assign multiple security groups to an instance, an instance can have hundreds of rules that apply. This might cause problems when you access the instance. Therefore, we recommend that you condense your rules as much as possible.

## Creating a Security Group

### To create a security group for EC2-Classic

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Security Groups**.
3. Click **Create Security Group**.
4. Specify a name and description for the security group. Select **No VPC** for **VPC**, and then click **Yes, Create**.

### To create a security group for EC2-VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Security Groups**.
3. Click **Create Security Group**.
4. Specify a name and description for the security group. Select the ID of your VPC for **VPC**, and then click **Yes, Create**.

## Describing Security Groups

### To view your security groups

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Security Groups**.
3. To view more information about a security group, including its rules, select it. The information is displayed in the details pane.

## Adding a Security Group Rule

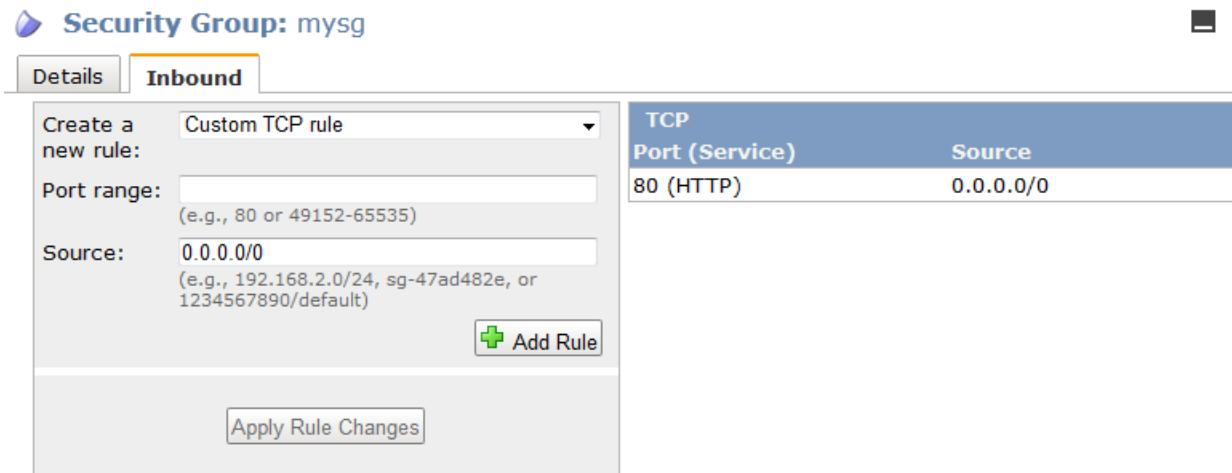
When you add a rule to a security group, the new rule is automatically applied to any instances in the group.

### To add a rule to a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Security Groups**.
3. Select a security group. The details pane displays information about the security group, plus tabs for working with its rules.
4. Select the **Inbound** tab and add one or more inbound rules.
  - a. Select an option from the **Create a new rule** list.
  - b. [Custom protocol rule only] Specify a port or port range.
  - c. In the **Source** field, specify one of the following:
    - An IP address range in CIDR notation (to allow access from that IP address range). Enter an IP address or range of addresses to limit access to one computer or a network, for example 203.0.113.5/32. However, for a Web server, you might enter 0.0.0.0/0 to allow all IP addresses over HTTP and HTTPS.

- The name or ID of a security group (to allow access based on that security group). If the security belongs to another AWS account, add the AWS account ID and a forward slash as a prefix (for example, 111122223333/OtherSecurityGroup).

d. Click **Add Rule**.



5. After adding all the rules for inbound traffic that you need, click **Apply Rule Changes**.

## Deleting a Security Group Rule

When you delete a rule from a security group, the change is automatically applied to any instances in the group.

### To delete a security group rule

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- In the navigation pane, click **Security Groups**.
- Select a security group.
- On the **Inbound** tab in the details pane, click the **Delete** button next to each rule to delete.

An asterisk appears on the **Inbound** tab to indicate that there are changes that have not been applied.

- Click **Apply Rule Changes**.

## Deleting a Security Group

The security group must not be assigned to any instances. You can't delete the default security group.

### To delete a security group

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- In the navigation pane, click **Security Groups**.
- Select a security group and click **Delete**.
- Click **Yes, Delete**.

## API and Command Overview

The following table summarizes the available commands and corresponding API actions for security groups.

Description	Command	API Action
Creates a security group.	<a href="#">ec2-create-group</a>	<a href="#">CreateSecurityGroup</a>
Adds one or more rules to a security group.	<a href="#">ec2-authorize</a>	<a href="#">AuthorizeSecurityGroupIngress</a> <a href="#">AuthorizeSecurityGroupEgress</a> (EC2-VPC only)
Describes one or more of your security groups.	<a href="#">ec2-describe-group</a>	<a href="#">DescribeSecurityGroups</a>
[EC2-VPC only] Modifies the security groups an instance is associated with.	<a href="#">ec2-modify-instance-attribute</a>	<a href="#">ModifyInstanceAttribute</a>
Removes one or more rules from a security group.	<a href="#">ec2-revoke</a>	<a href="#">RevokeSecurityGroupIngress</a>
Deletes a security group.	<a href="#">ec2-delete-group</a>	<a href="#">DeleteSecurityGroup</a>

## Controlling Access to Amazon EC2 Resources

When you sign up for an AWS account, you receive security credentials. These credentials grant you unlimited access to your AWS resources, such as your Amazon EC2 resources. You can use features of Amazon EC2 and AWS Identity and Access Management (IAM) to grant other users, services, and applications access to your Amazon EC2 resources without sharing the security credentials for your AWS account. You can choose to grant full access or limited access to your Amazon EC2 resources.

### Topics

- [Amazon EC2 Permission Attributes \(p. 421\)](#)
- [Introduction to IAM and Amazon EC2 \(p. 422\)](#)
- [IAM Policies for Amazon EC2 \(p. 423\)](#)
- [IAM Roles for Amazon EC2 \(p. 425\)](#)

## Amazon EC2 Permission Attributes

Your organization might have multiple AWS accounts. Amazon EC2 enables you to specify additional AWS accounts that can access your Amazon Machine Images (AMIs) and Amazon EBS snapshots. These permissions work at the AWS account level only; you can't restrict access to specific users within the specified AWS account. All users in the AWS account that you've specified can use the AMI or snapshot.

Each AMI has a `LaunchPermission` attribute that controls which AWS accounts can access the AMI. For more information, see [Sharing AMIs \(p. 62\)](#).

Each Amazon EBS snapshot has a `createVolumePermission` attribute that controls which AWS accounts can access the snapshot. For more information, see [Modifying Snapshot Permissions \(p. 498\)](#).

## Introduction to IAM and Amazon EC2

IAM enables you to do the following:

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

By using IAM with Amazon EC2, you can allow other users in your organization access to specific Amazon EC2 API actions.

## Creating an IAM Group and Users

### To create an IAM group and users

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. From the dashboard, click **Create a New Group of Users**.
3. On the **GROUP NAME** page, specify the name of the group under **Group Name**.
4. On the **PERMISSIONS** page, specify the policies for the group. You can select a policy template or create custom policies. For example, for Amazon EC2, one of the following policy templates might meet your needs:
  - Power User Access
  - Read Only Access
  - Amazon EC2 Full Access
  - Amazon EC2 Read Only Access
5. For more information about creating custom policies, see [IAM Policies for Amazon EC2 \(p. 423\)](#).
6. On the **USERS** page, specify one or more user names. If the users will use the CLI or API, select **Generate an access key for each User**. Click **Continue**.
7. If you had IAM generate access keys, click **Download Credentials** or **Show User Security Credentials** and save the access keys.
8. If the users will use the console, click **Users** in the navigation pane and do the following for each user:
  - a. Select the user.
  - b. Click the **Security Credentials** tab in the details pane.
  - c. Under **Sign-In Credentials**, click **Manage Password**.
  - d. In the **Manage Password** dialog box, select an option and click **Apply**.
  - e. Click **Download Credentials** or **Show User Security Credentials** and save the password.
9. Give each user his or her credentials (access keys and password); this enables them to access services based on the permissions you specified for the IAM group.

## Related Topics

For more information about IAM, see the following:

- [Identity and Access Management \(IAM\)](#)
- [IAM Getting Started Guide](#)
- [Using IAM](#)

## IAM Policies for Amazon EC2

By default, IAM users don't have permission to access the Amazon EC2 API. This means that they also can't use the Amazon EC2 console or CLI. To allow IAM users to use the features of Amazon EC2, you must create IAM policies that allow users to access the required Amazon EC2 API actions and resources, and then attach the policies to the IAM users or groups that require those permissions.

A policy is a JSON document that consists of one or more statements. Each statement is structured as follows:

```
{
 "Statement": [
 {
 "Effect": "effect",
 "Action": "action",
 "Resource": "arn",
 "Condition": {
 "condition": {
 "key": "value"
 }
 }
 }
]
}
```

The `effect` can be `Allow` or `Deny`. To learn about specifying `action`, see [Amazon EC2 API Actions \(p. 423\)](#). To learn about specifying `arn`, see [Amazon Resource Names \(ARNs\) for Amazon EC2 \(p. 423\)](#). To learn about specifying conditions for Amazon EC2, see [Amazon EC2 Policy Keys \(p. 424\)](#).

When you attach a policy to a user, that user is allowed to access the specified actions with the specified resources. When you attach a policy to a group, then every user in the group is allowed to access the specified actions and use them with the specified resources.

For more information about policies, see [Permissions and Policies](#) in the IAM documentation.

## Amazon EC2 API Actions

In an IAM policy, you can specify any of the Amazon EC2 API actions. Each action name must be prefixed with the lowercase string `ec2:`. For example: `ec2:RunInstances` or `ec2:CreateImage`. To specify all Amazon EC2 API actions, use `ec2:*`.

For a list of Amazon EC2 actions, see [Actions](#) in the [Amazon Elastic Compute Cloud API Reference](#).

## Amazon Resource Names (ARNs) for Amazon EC2

Amazon EC2 has no Amazon Resource Names (ARNs) because you can't specify particular Amazon EC2 resources in an IAM policy. When you write a policy to control access to Amazon EC2 API actions, specify the `*` wildcard as the resource.

For more information about ARNs, see [Amazon Resource Names \(ARN\) and AWS Service Namespaces](#) in the *Amazon Web Services General Reference*.

## Amazon EC2 Policy Keys

In an IAM policy, you can specify conditions that control when a policy is in effect. Each condition contains one or more key-value pairs. AWS defines conditions and keys, and individual services can define additional service-specific keys.

Amazon EC2 implements the AWS-wide policy keys, but no service-specific policy keys.

For a list of AWS-wide policy keys, see [Available Keys](#) in the IAM documentation.

## Example Policies for Amazon EC2

The following examples show simple policies that you could use to control the access that IAM users have to Amazon EC2. We specify the resource as "\*", because you can't specify particular Amazon EC2 resources in an IAM policy.

### Example 1: Allow users to list the Amazon EC2 resources that belong to the AWS account

The following policy allows users to access all Amazon EC2 API actions whose names begin with `Describe`. The users can't access any other API actions because users are denied access to the API actions by default.

```
{
 "Statement": [{
 "Effect": "Allow",
 "Action": "ec2:Describe*",
 "Resource": "*"
 }
]
```

### Example 2: Allow users to describe, launch, stop, start, and terminate instances

The following policy allows users to access the API actions specified in the `Action` element. The users can't access any other API actions because users are denied access to the API actions by default.

```
{
 "Statement": [{
 "Effect": "Allow",
 "Action": ["ec2:DescribeInstances", "ec2:DescribeImages",
 "ec2:DescribeKeyPairs", "ec2:DescribeSecurityGroups",
 "ec2:DescribeAvailabilityZones",
 "ec2:RunInstances", "ec2:TminateInstances",
 "ec2:StopInstances", "ec2:StartInstances"],
 "Resource": "*"
 }
]
```

## IAM Roles for Amazon EC2

Applications must sign their API requests with AWS credentials. Therefore, if you are an application developer, you need a strategy for managing credentials for your applications that run on Amazon EC2 instances. For example, you can securely distribute your AWS credentials to the instances, enabling the applications on those instances to use your credentials to sign requests, while protecting them from other users. However, it's challenging to securely distribute credentials to each instance, especially those that AWS creates on your behalf, such as Spot Instances or instances in Auto Scaling groups. You must also be able to update the credentials on each instance when you rotate your AWS credentials.

We designed IAM roles so that your applications can securely make API requests from your instances, without requiring you to manage the security credentials that the applications use. Instead of creating and distributing your AWS credentials, you can delegate API access using IAM roles as follows:

1. Create an IAM role.
2. Define which accounts or AWS services can assume the role.
3. Define which API actions and resources the application can access after assuming the role.
4. Specify the role when you launch your instances.
5. Have the application retrieve a set of temporary credentials and use them.

For example, you can use IAM roles to grant permissions to applications running on your instances that needs to access a bucket in Amazon S3.

You can specify permissions for IAM roles by creating a policy in JSON format. These are similar to the policies that you create for IAM users. If you make a change to a role, the change is propagated to all instances, simplifying credential management. For more information about creating and using IAM roles, see [Roles](#) in the IAM documentation.

## Retrieving Security Credentials from Instance Metadata

An application on the instance retrieves the security credentials provided by the role from the instance metadata item `iam/security-credentials/role-name`. The application uses the security credentials associated with the role to access resources using the permissions you've defined for the role. These security credentials are temporary and we rotate them automatically. We make new credentials available at least five minutes prior to the expiration of the old credentials.

### Warning

If you use services that use instance metadata with IAM roles, ensure that you don't expose your credentials when the services make HTTP calls on your behalf. The types of services that could expose your credentials include HTTP proxies, HTML/CSS validator services, and XML processors that support XML inclusion.

The following command retrieves the security credentials for an IAM role named `s3access`.

```
GET http://169.254.169.254/latest/meta-data/iam/security-credentials/s3access
{
 "Code" : "Success",
 "LastUpdated" : "2012-04-26T16:39:16Z",
 "Type" : "AWS-HMAC",
 "AccessKeyId" : "AKIAIOSFODNN7EXAMPLE",
 "SecretAccessKey" : "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
 "Token" : "token",
 "Expiration" : "2012-04-27T22:39:16Z"
}
```

For more information about instance metadata, see [Instance Metadata and User Data \(p. 365\)](#). For more information about temporary credentials, see [Using Temporary Security Credentials](#) in the IAM documentation.

## Launching an Instance with an IAM Role using the Console

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

### Important

After you create an IAM role, it may take several seconds for the permissions to propagate. If your first attempt to launch an instance with a role fails, wait a few seconds before trying again. For more information, see [Troubleshooting Working with Roles](#) in the IAM documentation.

### To launch an instance with an IAM role using the AWS Management Console

1. Create an IAM role.
  - a. Open the IAM console at <https://console.aws.amazon.com/iam/>.
  - b. In the navigation pane, click **Roles**, and then click **Create New Role**.
  - c. On the first **CONFIGURE ROLE** page, enter a name for the role and click **Continue**.
  - d. On the second **CONFIGURE ROLE** page, click the **Select** button for **Amazon EC2**.
  - e. On the **PERMISSIONS** page, specify the policies for the group. You can select a policy template or create custom policies. For example, for Amazon EC2, one of the following policy templates might meet your needs:
    - Power User Access
    - Read Only Access
    - Amazon EC2 Full Access
    - Amazon EC2 Read Only Access
- For more information about creating custom policies, see [IAM Policies for Amazon EC2 \(p. 423\)](#).
- f. Review the role information, edit the role as needed, and then click **Create Role**.
2. Launch an instance with the IAM role.
  - a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
  - b. On the dashboard, click **Launch Instance**.
  - c. If you're using the Quick Launch Wizard, do the following:
    - i. Provide the requested information on the first page, and then click **Continue**.
    - ii. Click **Edit details**.
    - iii. Select **Advanced Details** and select the IAM role you created from the **IAM Role** list.
    - iv. Click **Save details** and then click **Launch**.
  - d. If you're using the Classic Wizard, do the following:
    - i. Click the **Select** button for the AMI to use.
    - ii. Provide the requested information on the first **INSTANCE DETAILS** page and click **Continue**.
    - iii. On the second **INSTANCE DETAILS** page, select the IAM role you created from the **IAM Role** list and click **Continue**.
    - iv. Finish the wizard, clicking **Launch** on the final page.

3. If you are using the Amazon EC2 API actions in your application, retrieve the AWS security credentials made available on the instance and use them to sign the requests. Note that the AWS SDK does this for you.

```
GET http://169.254.169.254/latest/meta-data/iam/security-credentials/role_name
```

## Launching an Instance with an IAM Role using the CLI

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

### Important

After you create an IAM role, it may take several seconds for the permissions to propagate. If your first attempt to launch an instance with a role fails, wait a few seconds before trying again. For more information, see [Troubleshooting Working with Roles](#) in the IAM documentation.

### To launch an instance with an IAM role using the IAM and Amazon EC2 CLIs

1. The following example creates an IAM role named `s3access` with a policy that allows the role to access an S3 bucket, and an instance profile named `s3access`.

```
iam-rolecreate -r s3access -s ec2.amazonaws.com

iam-roleaddpolicy -r s3access -e Allow -a s3:/* -c /* -p s3star -o
{"Version": "2008-10-17", "Statement": [{"Effect": "Allow", "Action": ["s3:*"], "Resource": ["*"]}]}

iam-instanceprofilecreate -s s3access -r s3access
arn:aws:iam::111111111111:instance-profile/s3access
```

For more information, see [iam-rolecreate](#), [iam-roleaddpolicy](#), and [iam-instanceprofilecreate](#) in the *IAM Command Line Reference*.

2. Launch an instance using the instance profile. The following example shows a `t1.micro` instance being launched with the instance profile created in step 1.

```
ec2-run-instances -t t1.micro -p arn:aws:iam::111111111111:instance-profile/s3access -k key-pair -g 'Web Server' ami-e565ba8c
 RESERVATION r-11c62773 111111111111 sg-e7ddc68e
 INSTANCE i-9a6843fd ami-e565ba8c pending
key-pair 0 us-east-1e aki-88aa75e1 disabled

ec2-describe-instances
 RESERVATION r-11c62773 111111111111 sg-e7ddc68e
 INSTANCE i-9a6843fd ami-e565ba8c
 ec2-50-19-200-155.compute-1.amazonaws.com ip-10-28-28-186.ec2.in
ternal
 running key-pair 0 t1.micro 2012-04-26T16:29:25.000Z
 us-east-1e
 aki-88aa75e1 disabled
 AIPAJ6OQOSP4IRHXC16E4
```

For more information, see [ec2-run-instances](#) and [ec2-describe-instances](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

3. If you are using the Amazon EC2 API Actions in your application, retrieve the AWS security credentials made available on the instance and use them to sign the requests. Note that the AWS SDK does this for you.

```
GET http://169.254.169.254/latest/meta-data/iam/security-credentials/role_name
```

## Launching an Instance with an IAM Role using the AWS SDK

The AWS SDK documentation includes walkthroughs that show an application that uses security credentials from a IAM role to read an Amazon S3 bucket. For more information, see the following topics in the SDK documentation:

- [Using IAM Roles for EC2 Instances with the SDK for Java](#)
- [Using IAM Roles for EC2 Instances with the SDK for .NET](#)
- [Using IAM Roles for EC2 Instances with the SDK for PHP](#)
- [Using IAM Roles for EC2 Instances with the SDK for Ruby](#)

# Amazon EC2 and Amazon Virtual Private Cloud (VPC)

Amazon Virtual Private Cloud (Amazon VPC) enables you to define a virtual network in your own logically isolated area within the Amazon Web Services (AWS) cloud, known as a *virtual private cloud (VPC)*. You can launch your AWS resources, such as instances, into your VPC. Your VPC closely resembles a traditional network that you might operate in your own datacenter, with the benefits of using AWS's scalable infrastructure. You can configure your VPC; you can select its IP address range, create subnets, and configure route tables, network gateways, and security settings. You can connect instances in your VPC to the Internet. You can connect your VPC to your own corporate datacenter, making the AWS cloud an extension of your datacenter. To protect the resources in each subnet, you can use multiple layers of security, including security groups and network access control lists. For more information, see [Amazon Virtual Private Cloud User Guide](#).

## Benefits of Using a VPC

By launching your instances into a VPC instead of EC2-Classic, you gain the ability to:

- Assign static private IP addresses to your instances that persist across starts and stops
- Assign multiple IP addresses to your instances
- Define network interfaces, and attach one or more network interfaces to your instances
- Change security group membership for your instances while they're running
- Control the outbound traffic from your instances (egress filtering) in addition to controlling the inbound traffic to them (ingress filtering)
- Add an additional layer of access control to your instances in the form of network access control lists (ACL)
- Run your instances on single-tenant hardware

## Differences Between EC2-Classic and EC2-VPC

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

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

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
Public IP address	Your instance receives a public IP address.	Your instance launched in a default subnet receives a public IP address.	Your instance doesn't receive a public IP address.
Private IP address	Your instance receives a private IP address from the EC2-Classic, default VPC range each time it's started.	Your instance receives a static private IP address from the address range of your default VPC.	Your instance receives a static private IP address from the address range of your VPC.
Multiple IP addresses	You can assign a single IP address to your instance.	You can assign multiple IP addresses to your instance.	You can assign multiple IP addresses to your instance.
Elastic IP address	An EIP is disassociated from your instance when you stop it.	An EIP remains associated with your instance when you stop it.	An EIP remains associated with your instance when you stop it.
DNS hostnames	DNS hostnames are enabled by default.	DNS hostnames are enabled by default.	DNS hostnames are disabled by default.
Security group	A security group can reference security groups that belong to other AWS accounts.  You can create up to 500 security groups in each region.	A security group can reference security groups for your VPC only.  You can create up to 100 security groups per VPC.	A security group can reference security groups for your VPC only.  You can create up to 100 security groups per VPC.
Security group association	You must terminate your instance to change its security group.  You can assign an unlimited number of security groups to an instance.	You can change the security group of your running instance.  You can assign up to 5 security groups to an instance.	You can change the security group of your running instance.  You can assign up to 5 security groups to an instance.
Security group rules	You can add rules for inbound traffic only.  You can add up to 100 rules to a security group.	You can add rules for inbound and outbound traffic.  You can add up to 50 rules to a security group.	You can add rules for inbound and outbound traffic.  You can add up to 50 rules to a security group.

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
Tenancy	Your instance runs on shared hardware.	You can run your instance on shared hardware or single-tenant hardware.	You can run your instance on shared hardware or single-tenant hardware.

## Amazon VPC Documentation

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

Guide	Description
<a href="#">Amazon Virtual Private Cloud Getting Started Guide</a>	Provides a hands-on introduction to Amazon VPC.
<a href="#">Amazon Virtual Private Cloud User Guide</a>	Provides detailed information about how to use Amazon VPC.
<a href="#">Amazon Virtual Private Cloud Network Administrator Guide</a>	Helps network administrators configure your customer gateway.

## Amazon EC2 Instance IP Addressing

We provide your instances with IP addresses and DNS hostnames. These can vary depending on whether you launched the instance in the EC2-Classic platform or in a virtual private cloud (VPC).

For information about the EC2-Classic and EC2-VPC platforms, see [Supported Platforms \(p. 120\)](#). For information about Amazon VPC, see [What is Amazon VPC?](#) in the *Amazon Virtual Private Cloud User Guide*.

### Topics

- [Private Addresses and Internal DNS Hostnames \(p. 430\)](#)
- [Public IP Addresses and External DNS Hostnames \(p. 431\)](#)
- [Differences Between EC2-Classic and EC2-VPC \(p. 431\)](#)
- [Determining Your Public and Private IP Addresses \(p. 432\)](#)
- [Multiple IP Addresses \(p. 433\)](#)

## Private Addresses and Internal DNS Hostnames

You can use private IP addresses and internal DNS hostnames for communication between instances in the same network (EC2-Classic or a VPC).

When you launch an instance, we allocate a private IP address for the instance using DHCP. Private IP addresses are not reachable from the Internet. For more information about private IP addresses, see [RFC 1918](#).

Each instance that you launch into a VPC has a default network interface. The network interface specifies the primary private IP address for the instance. If you don't select a primary private IP address, we select an available IP address in the subnet's range. You can specify additional private IP addresses, known as secondary private IP addresses. Unlike primary private IP addresses, secondary private IP addresses can be reassigned from one instance to another. For more information, see [Multiple IP Addresses \(p. 433\)](#).

Each instance is provided an internal DNS hostname that resolves to the private IP address of the instance in EC2-Classic or your VPC. We can't resolve the DNS hostname outside the network that the instance is in.

For instances launched in EC2-Classic, a private IP address is associated with the instance until it is stopped or terminated.

For instances launched in a VPC, a private IP address remains associated with the network interface when the instance is stopped and restarted, and is released when the instance is terminated.

## Public IP Addresses and External DNS Hostnames

You can use public IP addresses and external DNS hostnames for communication between your instances and the Internet or other AWS products, such as Amazon Simple Storage Service (Amazon S3).

When you launch an instance in EC2-Classic or a default subnet, we allocate a public IP address for the instance. Public IP addresses are reachable from the Internet. A public IP address is associated with an instance until it is stopped or terminated. If you require a persistent public IP address that can be assigned to and removed from instances as necessary, use an Elastic IP address. For more information, see [Elastic IP Addresses \(EIP\) \(p. 438\)](#).

We don't allocate a public IP address for an instance that you launch in a nondefault subnet. Therefore, if you want an instance in a nondefault subnet to communicate with the Internet, you must associate an Elastic IP address with the primary or any secondary private IP address assigned to the network interface for the instance.

We provide each instance that has a public IP address with an external DNS hostname. We resolve an external DNS hostname to the public IP address of the instance outside the network of the instance, and to the private IP address of the instance from within the network of the instance. If your instance is in a VPC and you assign it an Elastic IP address, it receives a DNS hostname if DNS hostnames are enabled. For more information, see [Using DNS with Your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

The private IP address and public IP address for an instance are directly mapped to each other through network address translation (NAT). For more information about NAT, see [RFC 1631: The IP Network Address Translator \(NAT\)](#).

**Note**

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

## Differences Between EC2-Classic and EC2-VPC

The following table summarizes the differences between IP addresses for instances launched in EC2-Classic, instances launched in a default subnet, and instances launched in a nondefault subnet.

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
Public IP address	Your instance receives a public IP address.	Your instance launched in a default subnet receives a public IP address.	Your instance doesn't receive a public IP address.
Private IP address	Your instance receives a private IP address from the EC2-Classic range each time it's started.	Your instance receives a static private IP address from the address range of your default VPC.	Your instance receives a static private IP address from the address range of your VPC.

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
Multiple IP addresses	We select a single private IP address for your instance; multiple IP addresses are not supported.	You can assign multiple private IP addresses to your instance.	You can assign multiple private IP addresses to your instance.
Network interfaces	IP addresses are associated with the instance; network interfaces aren't supported.	IP addresses are associated with a network interface. Each instance has one or more network interfaces.	IP addresses are associated with a network interface. Each instance has one or more network interfaces.
Elastic IP address	An EIP is disassociated from your instance when you stop it.	An EIP remains associated with your instance when you stop it.	An EIP remains associated with your instance when you stop it.
DNS hostnames	DNS hostnames are enabled by default.	DNS hostnames are enabled by default.	DNS hostnames are disabled by default.

## Determining Your Public and Private IP Addresses

You can determine the public and private IP addresses of your instances using the Amazon EC2 console.

### To determine your instance's IP addresses using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Instances** in the navigation pane.
3. Select an instance. The console displays information about the instance in the lower pane.
4. Get the public IP address from the **Public DNS** field.
5. Get the private IP address from the **Private IP** field.

You can also determine the public and private IP addresses of your instances using instance metadata. For more information about instance metadata, see [Instance Metadata and User Data \(p. 365\)](#).

### To determine your instance's IP addresses using instance metadata

1. Connect to the instance.
2. Use the following command to access the private IP address:

```
GET http://169.254.169.254/latest/meta-data/local-ipv4
```

3. Use the following command to access the public IP address:

```
GET http://169.254.169.254/latest/meta-data/public-ipv4
```

## Multiple IP Addresses

In EC2-VPC, you can specify multiple private IP addresses for your instances. The number of network interfaces and private IP addresses that you can specify for an instance depends on the instance type. For more information, see [Private IP Addresses Per ENI Per Instance Type \(p. 91\)](#).

It can be useful to assign multiple private IP addresses to an instance in your VPC to do the following:

- Host multiple websites on a single server by using multiple SSL certificates on a single server and associating each certificate with a specific IP address.
- Operate network appliances, such as firewalls or load balancers, that have multiple private IP addresses for each network interface.
- Redirect internal traffic to a standby instance in case your instance fails, by reassigning the secondary private IP address to the standby instance.

### Topics

- [How Multiple IP Addresses Work \(p. 433\)](#)
- [Assigning a Secondary Private IP Address \(p. 434\)](#)
- [Configuring the Operating System on Your Instance to Recognize the Secondary Private IP Address \(p. 436\)](#)
- [Assigning an Elastic IP Address to the Secondary Private IP Address \(p. 437\)](#)
- [Viewing Your Secondary Private IP Addresses \(p. 437\)](#)
- [Unassigning a Secondary Private IP Address \(p. 437\)](#)

## How Multiple IP Addresses Work

The following list explains how multiple IP addresses work with network interfaces:

- You can assign a secondary private IP address to any network interface. The network interface can be attached to or detached from the instance.
- You must choose a secondary private IP address that's in the CIDR block range of the subnet for the network interface.
- Security groups apply to network interfaces, not to IP addresses. Therefore, IP addresses are subject to the security group of the network interface in which they're specified.
- Secondary private IP addresses can be assigned and unassigned to elastic network interfaces attached to running or stopped instances.
- Secondary private IP addresses that are assigned to a network interface can be reassigned to another one if you explicitly allow it.
- When assigning multiple secondary private IP addresses to a network interface using the command line tools or API, the entire operation fails if one of the secondary private IP addresses can't be assigned.
- Primary private IP addresses, secondary private IP addresses, and any associated Elastic IP addresses remain with the network interface when it is detached from an instance or attached to another instance.
- Although you can't move the primary network interface from an instance, you can reassign the secondary private IP address of the primary network interface to another network interface.
- You can move any additional network interface from one instance to another.

The following list explains how multiple IP addresses work with Elastic IP addresses:

- Each private IP address can be associated with a single Elastic IP address, and vice versa.
- When a secondary private IP address is reassigned to another interface, the secondary private IP address retains its association with an Elastic IP address.

- When a secondary private IP address is unassigned from an interface, an associated Elastic IP address is automatically disassociated from the secondary private IP address.

## Assigning a Secondary Private IP Address

You can assign the secondary private IP address to the network interface for an instance as you launch the instance, or after the instance is running.

### To assign a secondary private IP address when launching an instance in EC2-VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click the **Launch Instance** button.
3. On the **Create a New Instance** page, click **Classic Wizard**, and then click **Continue**.
4. On the **CHOOSE AN AMI** page, the **Quick Start** tab displays a list of basic configurations called Amazon Machine Images (AMI). Choose an AMI and click its **Select** button.
5. On the **INSTANCE DETAILS** page, set the number and type of instance to launch. You can select a subnet or let us choose one. Confirm your selections, and then click **Continue**.
6. On the next **INSTANCE DETAILS** page, under **Advanced Instance Options**, do the following, and then click **Continue**:
  - a. Select the number of network interfaces to attach to the instance. The console enables you specify up to 2 network interfaces when you launch an instance. After you launch the instance, click **Network Interfaces** in the navigation pane to add additional network interfaces. The total number of network interfaces that you can attach varies by instance type. For more information, see [Private IP Addresses Per ENI Per Instance Type \(p. 91\)](#).
  - b. For each network interface, you can specify a primary private IP address, and one or more secondary private IP addresses. For this example, however, accept the IP address that we automatically assign.
  - c. Beside **Secondary IP Addresses**, click **Add**, and then enter a private IP address in the subnet range, or accept the default, `auto-assign`, to let us select an address.

#### Important

After you have added a secondary private IP address to a network interface, you must connect to the instance and configure the secondary private IP address on the instance itself. For more information, see [Configuring the Operating System on Your Instance to Recognize the Secondary Private IP Address \(p. 436\)](#).

## Request Instances Wizard

CHOOSE AN AMI

INSTANCE DETAILS

CREATE KEY PAIR

CONFIGURE FIREWALL

REVIEW

**Number of Instances:** 1

**Availability Zone:** No Preference

### Advanced Instance Options

Here you can choose a specific **kernel** or **RAM disk** to use with your instances. You can also choose **Detailed Monitoring** or enter data that will be available from your instances once they launch.

**Kernel ID:**

Use Default ▾

**RAM Disk ID:**

Use Default

**Monitoring:**

Enable CloudWatch detailed monitoring for this instance  
(additional charges will apply)

**User Data:**

as text

as file

base64 encoded

**Termination Protection:**

Prevention against accidental termination.

**Shutdown Behavior:**

Stop

**IAM Role:**  None ▾

**Tenancy:**

Default

**Number of Network Interfaces:** 1 ▾

eth0

Network Interface: New Interface

Secondary IP

Addresses: auto-assign

Subnet: subnet-cd8a35a4 (10.0.1.0/24) ▾

Add

IP Address: auto-assign

[◀ Back](#)

[Continue](#) ▶

7. On the next **INSTANCE DETAILS** page, you can specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume), and then click **Continue**.
8. On the final **INSTANCE DETAILS** page, specify tags for the instance, such as a user-friendly name, and then click **Continue**.
9. On the **CREATE KEY PAIR** page, select an existing key pair or create a new one. If you create a new key pair, you must download it before you can click **Continue**.

If you're new to Amazon EC2 and haven't created any key pairs, the wizard prompts you to create one.

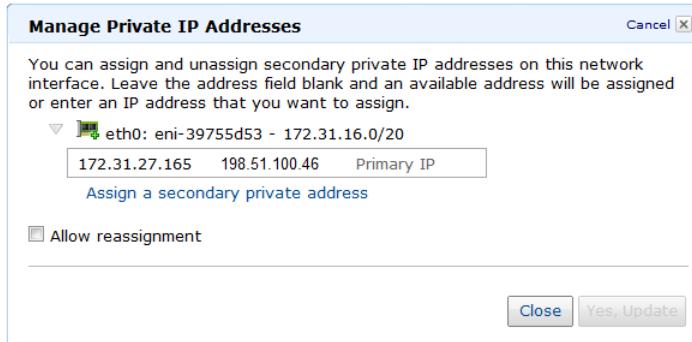
10. On the **CONFIGURE FIREWALL** page, select an existing security group for the primary network interface, or create a new one, and then click **Continue**.

The security group for the additional network interface was previously selected when the ENI was created.

11. On the **REVIEW** page, review the settings, and then click **Launch** to begin launching your instance.

### To assign a secondary private IP to an existing instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, do one of the following:
  - Click **Instances**, and then right-click the instance.
  - Click **Network Interfaces**, and then right-click the network interface attached to the instance.
3. Select **Manage Private IP Addresses**.
4. In the **Manage Private IP Addresses** dialog box, do the following:
  - a. Click **Assign a secondary private address**.
  - b. In the **Address** field, enter a specific IP address that's within the subnet range for the instance, or leave the field blank and we'll select an IP address for you.
  - c. (Optional) Select **Allow reassignment** to allow the secondary private IP address to be reassigned if it is already assigned to another network interface.
  - d. Click **Yes, Update**, and then click **Close**.



## Configuring the Operating System on Your Instance to Recognize the Secondary Private IP Address

After you assign a secondary private IP address to your instance, you need to configure the operating system on your instance to recognize the secondary private IP address.

If you are using Amazon Linux, the `ec2-net-utils` package takes care of this step for you. It configures additional network interfaces attached while the instance is running, refreshes secondary IP addresses during DHCP lease renewal, and updates the related routing rules. If you require manual control over network configuration, you can remove the `ec2-net-utils` package.

If you are using another Linux distribution, see the documentation for your Linux distribution. Search for information about configuring additional network interfaces and secondary IP addresses. If the instance has two or more interfaces on the same subnet, search for information about using routing rules to work around asymmetric routing.

For information about configuring a Windows instance, see [Configuring a Secondary Private IP Address for Your Windows Instance](#) in the *Amazon Elastic Compute Cloud Microsoft Windows Guide*.

## Assigning an Elastic IP Address to the Secondary Private IP Address

### To assign an EIP to a secondary private IP address in EC2-VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Elastic IPs** in the navigation pane.
3. Right-click the IP address, and then click **Associate**.
4. In the **Associate Address** dialog box, select the network interface from the **Network Interface** drop-down list, and then select the secondary IP address from the **Private IP address** drop-down list.
5. Click **Yes, Associate**.

## Viewing Your Secondary Private IP Addresses

### To view the private IP addresses assigned to a network interface in EC2-VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Network Interface** in the navigation pane.
3. Select the network interface whose private IP addresses you want to view.
4. On the **Details** tab in the details pane, check the **Primary IP** and **Secondary Private IPs** fields for the primary private IP address and any secondary private IP addresses assigned to the network interface.

### To view the private IP addresses assigned to an instance

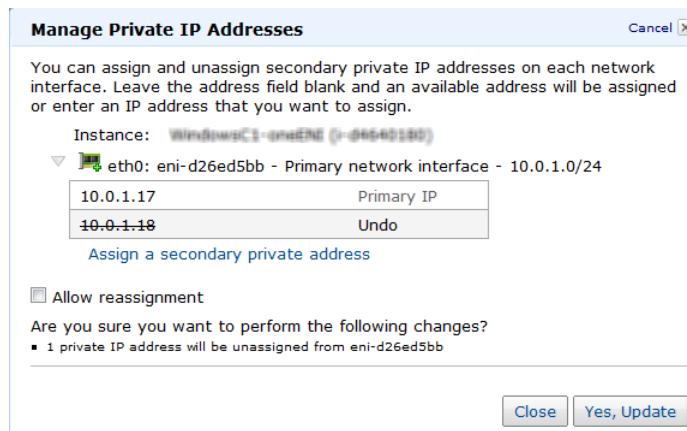
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Instances** in the navigation pane.
3. Select the instance whose private IP addresses you want to view.
4. On the **Description** tab in the details pane, check the **Primary IPs** and **Secondary Private IPs** fields for the primary private IP address and any secondary private IP addresses assigned to the instance through its network interface.

## Unassigning a Secondary Private IP Address

If you no longer require a secondary private IP address, you can unassign it from the instance or the network interface. When a secondary private IP address is unassigned from an elastic network interface, the Elastic IP address (if it exists) is also disassociated.

### To unassign a secondary private IP address from an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Instances** in the navigation pane.
3. Right-click an instance, and then click **Manage Private IP Addresses**.
4. In the **Manage Private IP Addresses** dialog box, beside the secondary private IP address to unassign, click **Unassign**.
5. Click **Yes, Update**, and then click **Close**.



### To unassign a secondary private IP address from a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Network Interface** in the navigation pane.
3. Right-click an instance, and click **Manage Private IP Addresses**.
4. In the **Manage Private IP Addresses** dialog box, beside the secondary private IP address to unassign, click **Unassign**.
5. Click **Yes, Update**, and then click **Close**.



## Elastic IP Addresses (EIP)

An *Elastic IP address (EIP)* is a static IP address designed for dynamic cloud computing. With an EIP, you can mask the failure of an instance by rapidly remapping the address to another instance. Your EIP is associated with your AWS account, not a particular instance, and it remains associated with your account until you choose to explicitly release it.

There's one pool of EIPs for use with the EC2-Classic platform and another for use with your VPC. You can't associate an EIP that you allocated for use with a VPC with an instance in EC2-Classic, and vice-versa. For more information about EC2-Classic and EC2-VPC, see [Supported Platforms \(p. 120\)](#).

### Topics

- [Elastic IP Addresses in EC2-Classic \(p. 439\)](#)
- [Elastic IP Addresses in a VPC \(p. 439\)](#)
- [Differences Between EC2-Classic and EC2-VPC \(p. 440\)](#)
- [Allocating an Elastic IP Address \(p. 440\)](#)
- [Describing Your Elastic IP Addresses \(p. 440\)](#)
- [Associating an Elastic IP Address with a Running Instance \(p. 441\)](#)
- [Associating an Elastic IP Address with a Different Running Instance \(p. 441\)](#)
- [Releasing an Elastic IP Address \(p. 441\)](#)
- [Using Reverse DNS for Email Applications \(p. 441\)](#)
- [API and CLI Overview \(p. 442\)](#)
- [Elastic IP Address Limit \(p. 442\)](#)

## Elastic IP Addresses in EC2-Classic

By default, we assign each instance in EC2-Classic two IP addresses at launch: a private IP address and a public IP address that is mapped to the private IP address through network address translation (NAT).

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

When you associate an EIP with an instance, its current public IP address is released to the EC2-Classic public IP address pool. If you disassociate an EIP from the instance, the instance is automatically assigned a new public IP address within a few minutes. In addition, stopping the instance also disassociates the EIP from it.

To ensure efficient use of EIPs, we impose a small hourly charge when they are not associated with a running instance, or when they are associated with a stopped instance.

## Elastic IP Addresses in a VPC

We assign each instance in a default VPC two IP addresses at launch: a private IP address and a public IP address that is mapped to the private IP address through network address translation (NAT).

We assign each instance in a nondefault VPC only a private IP address. Therefore, if you want an instance in a nondefault VPC to communicate with the Internet, you must allocate an Elastic IP address for use with a VPC, and then associate that EIP with the elastic network interface (ENI) attached to the instance.

When you associate an EIP with an instance in a default VPC, its current public IP address is released to the EC2-VPC public IP address pool. If you disassociate an EIP from the instance, the instance is automatically assigned a new public IP address within a few minutes. The EIP remains associated with the instance when you stop it.

To ensure efficient use of EIPs, we impose a small hourly charge when they are not associated with a running instance, or when they are associated with a stopped instance or an unattached network interface. While your instance is running, you are not charged for one EIP associated with the instance, but you are charged for any additional EIPs associated with the instance.

For information about using an EIP with an instance in a VPC, see [Elastic IP Addresses](#) in the *Amazon Virtual Private Cloud User Guide*.

## Differences Between EC2-Classic and EC2-VPC

The following table lists the differences between EIPs on EC2-Classic and EC2-VPC.

Characteristic	EC2-Classic	EC2-VPC
Allocation	When you allocate an EIP, it's for use only in EC2-Classic.	When you allocate an EIP, it's for use only in a VPC.
Association	You associate an EIP with an instance.	An EIP is a property of an elastic network interface (ENI). You can associate an EIP with an instance by updating the ENI attached to the instance. For more information, see <a href="#">Elastic Network Interfaces (ENI) (p. 442)</a> .
Reassociation	If you try to associate an EIP that's already associated with another instance, the address is automatically associated with the new instance.	If you try to associate an EIP that's already associated with another instance, it succeeds only if you allowed reassociation.
Instance stop	If you stop an instance, its EIP is disassociated, and you must re-associate the EIP when you restart the instance.	If you stop an instance, its EIP remains associated.
Multiple IP	Instances support only a single private IP address and a corresponding EIP.	Instances support multiple IP addresses, and each one can have a corresponding EIP. For more information, see <a href="#">Multiple IP Addresses (p. 433)</a> .

## Allocating an Elastic IP Address

### To allocate an EIP for use with EC2-Classic

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Elastic IPs** in the navigation pane.
3. Click **Allocate New Address**.
4. Select **EC2** from the **EIP** list, and then click **Yes, Allocate**.

## Describing Your Elastic IP Addresses

### To view your Elastic IP addresses

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Elastic IPs** in the navigation pane.
3. To filter the displayed list, start typing part of the EIP or the ID of the instance to which it is assigned in the search box.

## Associating an Elastic IP Address with a Running Instance

### To associate an Elastic IP address with a running instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Elastic IPs** in the navigation pane.
3. Select an EIP and click **Associate Address**.
4. In the **Associate Address** dialog box, select the instance from the **Instance** list box and click **Yes, Associate**.

## Associating an Elastic IP Address with a Different Running Instance

### To reassocciate an Elastic IP address

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Elastic IPs** in the navigation pane.
3. Select the EIP, and then click the **Disassociate** button.
4. Click **Yes, Disassociate** when prompted.
5. Select the EIP, and then click **Associate**.
6. In the **Associate Address** dialog box, select the new instance from the **Instance ID** list, and then click **Yes, Associate**.

## Releasing an Elastic IP Address

If you no longer need an EIP, we recommend that you release it (the address must not be associated with an instance). You incur charges for any EIP that's allocated for use with EC2-Classic but not associated with an instance.

### To release an Elastic IP address

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Elastic IPs** in the navigation pane.
3. Select the Elastic IP address, and then click the **Release Address** button.

## Using Reverse DNS for Email Applications

If you intend to send email to third parties from an instance, we suggest you provision one or more Elastic IP addresses and provide them to us in the [Request to Remove Email Sending Limitations form](#). AWS works with ISPs and Internet anti-spam organizations (such as Spamhaus) to reduce the chance that your email sent from these addresses will be flagged as spam.

In addition, assigning a static reverse DNS record to your Elastic IP address used to send email can help avoid having email flagged as spam by some anti-spam organizations. You can provide us with a reverse DNS record to associate with your addresses through the aforementioned form. Note that a corresponding forward DNS record (A Record) pointing to your Elastic IP address must exist before we can create your reverse DNS record.

## API and CLI Overview

The following table summarizes the available Elastic IP address command line interface (CLI) and corresponding API actions.

Description	CLI Command	API Action
Acquires an Elastic IP address for your account.	<a href="#">ec2-allocate-address</a>	<a href="#">AllocateAddress</a>
Associates an Elastic IP address with an instance or a network interface.	<a href="#">ec2-associate-address</a>	<a href="#">AssociateAddress</a>
Describes one or more of your Elastic IP addresses.	<a href="#">ec2-describe-addresses</a>	<a href="#">DescribeAddresses</a>
Disassociates an Elastic IP address from the instance or network interface it's associated with.	<a href="#">ec2-disassociate-address</a>	<a href="#">DisassociateAddress</a>
Releases an Elastic IP address allocated to your account.	<a href="#">ec2-release-address</a>	<a href="#">ReleaseAddress</a>

## Elastic IP Address Limit

By default, all AWS accounts are limited to 5 EIPs, because public (IPv4) Internet addresses are a scarce public resource. We strongly encourage you to use an EIP primarily for load balancing use cases, and use DNS hostnames for all other inter-node communication.

If you feel your architecture warrants additional EIPs, please complete the [Amazon EC2 Elastic IP Address Request Form](#). We will ask you to describe your use case so that we can understand your need for additional addresses.

## Elastic Network Interfaces (ENI)

An elastic network interface (ENI) is a virtual network interface that you can attach to an instance in a VPC. An ENI can include the following attributes:

- a primary private IP address
- one or more secondary private IP addresses
- an Elastic IP address
- a MAC address
- one or more associated security groups
- a source/destination check flag
- a description

You can create a network interface, attach it to an instance, detach it from an instance, and attach it to another instance. The attributes of a network interface follow the network interface as it is attached or detached from an instance and reattached to another instance. When you move a network interface from one instance to another, network traffic is redirected to the new instance.

Each instance in a VPC has a default network interface. The default network interface has a primary private IP address in the IP address range of its VPC. You can create and attach additional network interfaces. The maximum number of network interfaces that you can use varies by instance type. For more information, see [Private IP Addresses Per ENI Per Instance Type \(p. 91\)](#).

Attaching multiple network interfaces to an instance is useful when you want to:

- Create a management network.
- Use network and security appliances in your VPC.
- Create dual-homed instances with workloads/roles on distinct subnets.
- Create a low-budget, high-availability solution.

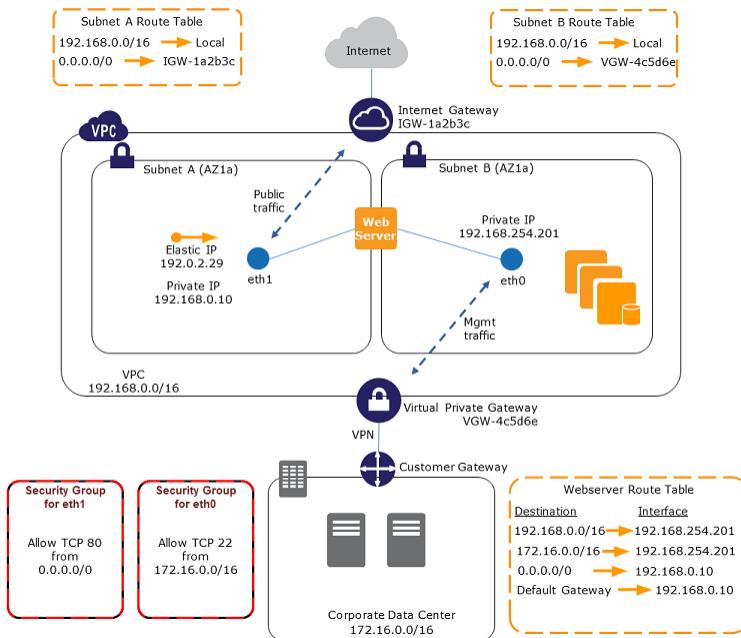
### Topics

- [Creating a Management Network \(p. 443\)](#)
- [Use Network and Security Appliances in Your VPC \(p. 444\)](#)
- [Creating Dual-homed Instances with Workloads/Roles on Distinct Subnets \(p. 444\)](#)
- [Create a Low Budget High Availability Solution \(p. 444\)](#)
- [Best Practices for Configuring Network Interfaces \(p. 445\)](#)
- [Creating a Network Interface \(p. 445\)](#)
- [Deleting a Network Interface \(p. 446\)](#)
- [Viewing Details about a Network Interface \(p. 446\)](#)
- [Attaching a Network Interface When Launching an Instance \(p. 446\)](#)
- [Attaching a Network Interface to a Stopped or Running Instance \(p. 448\)](#)
- [Detaching a Network Interface from an Instance \(p. 448\)](#)
- [Changing the Security Group of a Network Interface \(p. 449\)](#)
- [Changing the Source/Destination Checking of a Network Interface \(p. 449\)](#)
- [Associating an Elastic IP Address with a Network Interface \(p. 450\)](#)
- [Disassociating an Elastic IP Address from a Network Interface \(p. 450\)](#)
- [Changing Termination Behavior for a Network Interface \(p. 451\)](#)
- [Adding or Editing a Description for a Network Interface \(p. 451\)](#)
- [Adding or Editing Tags for a Network Interface \(p. 452\)](#)
- [API and Command Overview \(p. 452\)](#)

## Creating a Management Network

You can create a management network using network interfaces. In this scenario, the secondary network interface on the instance handles public-facing traffic and the primary network interface handles back-end management traffic and is connected to a separate subnet in your VPC that has more restrictive access controls. The public facing interface, which may or may not be behind a load balancer, has an associated security group that allows access to the server from the Internet (for example, allow TCP port 80 and 443 from 0.0.0.0/0, or from the load balancer) while the private facing interface has an associated security group allowing SSH access only from an allowed range of IP addresses either within the VPC or from the Internet, a private subnet within the VPC or a virtual private gateway.

To ensure failover capabilities, consider using a secondary private IP for incoming traffic on a network interface. In the event of an instance failure, you can move the interface and/or secondary private IP address to a standby instance.



## Use Network and Security Appliances in Your VPC

Some network and security appliances, such as load balancers, network address translation (NAT) servers, and proxy servers prefer to be configured with multiple network interfaces. You can create and attach secondary network interfaces to instances in a VPC that are running these types of applications and configure the additional interfaces with their own public and private IP addresses, security groups, and source/destination checking.

## Creating Dual-homed Instances with Workloads/Roles on Distinct Subnets

You can place a network interface on each of your web servers that connects to a mid-tier network where an application server resides. The application server can also be dual-homed to a back-end network (subnet) where the database server resides. Instead of routing network packets through the dual-homed instances, each dual-homed instance receives and processes requests on the front end, initiates a connection to the back end, and then sends requests to the servers on the back-end network.

## Create a Low Budget High Availability Solution

If one of your instances serving a particular function fails, its network interface can be attached to a replacement or hot standby instance pre-configured for the same role in order to rapidly recover the service. For example, you can use an ENI as your primary or secondary network interface to a critical service such as a database instance or a NAT instance. If the instance fails, you (or more likely, the code running on your behalf) can attach the ENI to a hot standby instance. Because the interface maintains its private IP addresses, Elastic IP addresses, and MAC address, network traffic will begin flowing to the standby instance as soon as you attach the ENI to the replacement instance. Users will experience a brief loss of connectivity between the time the instance fails and the time that the ENI is attached to the standby instance, but no changes to the VPC route table or your DNS server are required.

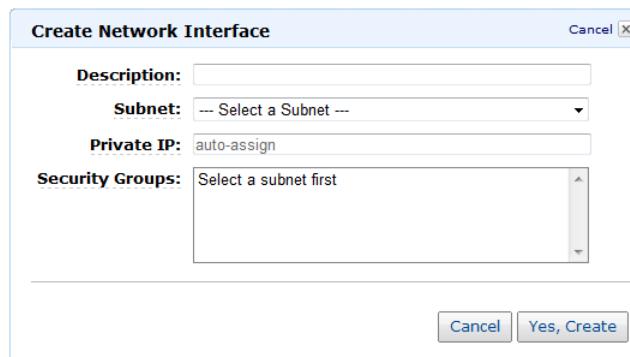
## Best Practices for Configuring Network Interfaces

- You can attach a network interface to an instance when it's running (hot attach), when it's stopped (warm attach), or when the instance is being launched (cold attach).
- You can detach secondary (eth-n) network interfaces when the instance is running or stopped. However, you cannot detach the primary (eth0) interface.
- You can attach a network interface in one subnet to an instance in another subnet in the same VPC, however, both the network interface and the instance must reside in the same Availability Zone.
- When launching an instance from the CLI or API, you can specify the network interfaces to attach to the instance for both the primary (eth0) and additional network interfaces.
- Launching an instance with multiple network interfaces automatically configures interfaces, private IP addresses, and route tables on the operating system of the instance. A warm or hot attach of an additional network interface may require you to manually bring up the second interface, configure the private IP address, and modify the route table accordingly. (Instances running Microsoft Windows Server or Amazon Linux automatically recognize the warm or hot attach and configure themselves.)
- Attaching another network interface to an instance is not a method to increase or double the network bandwidth to or from the dual-homed instance.

## Creating a Network Interface

### To create a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Network Interfaces** in the navigation pane.
3. Click **Create Network Interface**.
4. In the **Create Network Interface** dialog box, provide the following information for the network interface, and then click **Yes, Create**.
  - a. In **Description**, enter a descriptive name.
  - b. In **Subnet**, select the subnet. Note that you can't move the network interface to another subnet after it's created.
  - c. In **Private IP**, enter the primary private IP address. If you don't specify an IP address, we'll select an available IP address from within the selected subnet.
  - d. In **Security Groups**, select one or more security groups.

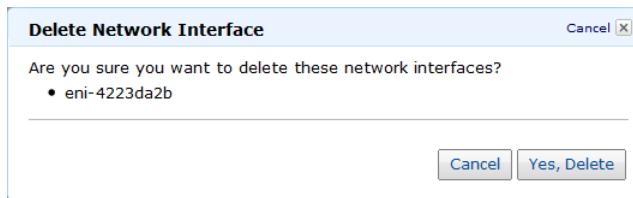


## Deleting a Network Interface

You must first detach a network interface from an instance before you can delete it. Deleting a network interface releases all attributes associated with the network interface and releases any private IP addresses or Elastic IP addresses to be used by another instance.

### To delete a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Network Interfaces** in the navigation pane.
3. Select a network interface, and then click the **Delete** button.
4. In the **Delete Network Interface** dialog box, click **Yes, Delete**.



## Viewing Details about a Network Interface

### To view details about a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Network Interfaces** in the navigation pane.
3. Select the network interface.
4. View the details on the **Details** tab.

A screenshot of the 'Network Interface' details page. It shows a summary table with tabs for 'Details' and 'Tags'. The 'Details' tab is selected. The table includes columns for Network Interface ID, VPC, MAC Address, Security Groups, Status, Private DNS, Source/Dest. Check, Instance, and Device Index. To the right, a detailed table lists Subnet, Zone, Description, Owner, Private IP, Secondary Private IPs, Attachment ID, Attachment Owner, and Attachment Status.

Network Interface ID:	eni-a66ed5cf	Subnet:	subnet-cd8a35a4
VPC:	vpc-f28a359b	Zone:	ap-southeast-1b
MAC Address:	02:78:d7:00:8a:1e	Description:	Primary network interface
Security Groups:	quick-start-1	Owner:	053230519467
Status:	in-use	Private IP:	10.0.1.233
Private DNS:	-	Secondary Private IPs:	10.0.1.20
Source/Dest. Check:	enabled	Attachment ID:	eni-attach-a99c57c0
Instance:	i-886401dc	Attachment Owner:	053230519467
Device Index:	0	Attachment Status:	attached

## Attaching a Network Interface When Launching an Instance

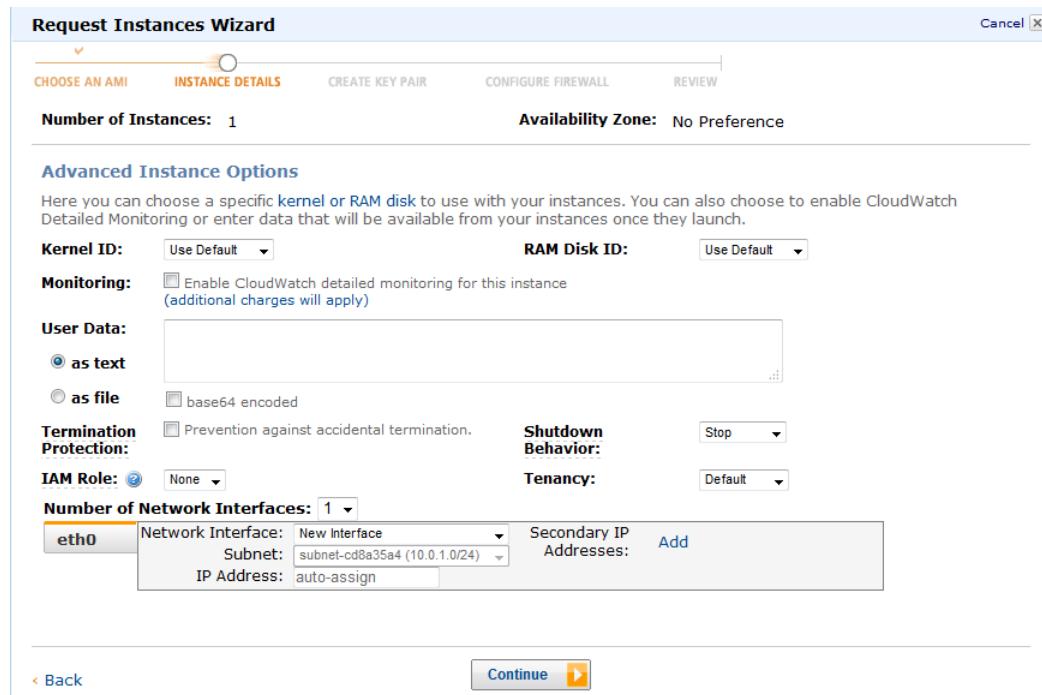
You can attach an additional network interface, designated as eth1-*n*, to an instance when you launch it into a VPC.

**Note**

If an error occurs when attaching a network interface to your instance, this causes the instance launch to fail.

**To attach a network interface when launching an instance**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Launch Instance**.
3. On the **Create a New Instance** page, click **Classic Wizard**, and then click **Continue**.
4. On the **CHOOSE AN AMI** page, the **Quick Start** tab displays a list of basic configurations called Amazon Machine Images (AMI). Choose an AMI and click its **Select** button.
5. On the **INSTANCE DETAILS** page, set the number and type of instance to launch. You can select a subnet or let us choose one. Confirm your selections, and then click **Continue**.
6. On the next **INSTANCE DETAILS** page, under **Advanced Instance Options**, select the number of network interfaces to attach to the instance. The console enables you specify up to 2 network interfaces when you launch an instance. After you launch the instance, click **Network Interfaces** in the navigation pane to add additional network interfaces. The total number of network interfaces that you can attach varies by instance type. For more information, see [Private IP Addresses Per ENI Per Instance Type \(p. 91\)](#). You can also enter an IP address for the primary network interface (eth0). When you've finished, click **Continue**.



7. On the **CREATE KEY PAIR** page, select an existing key pair or create a new one. If you create a new key pair, you must download it before you can click **Continue**.  
If you're new to Amazon EC2 and haven't created any key pairs, the wizard prompts you to create one.
8. On the **CONFIGURE FIREWALL** page, select an existing security group for the primary network interface or create a new one, and then click **Continue**.

The security group for the additional network interface was previously selected when the network interface was created.

9. On the **REVIEW** page, details about the primary and additional network interface are displayed. Review the settings, and then click **Launch**.

## Attaching a Network Interface to a Stopped or Running Instance

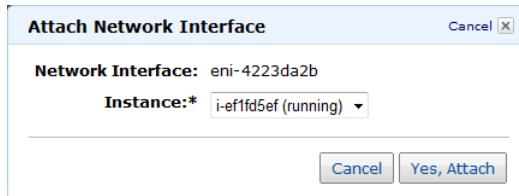
You can attach a network interface to any of your stopped or running instances in your VPC from either the **Instances** page or the **Network Interfaces** page of the EC2 console.

### To attach a network interface to a stopped or running instance using Instances

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Instances** in the navigation pane.
3. Right-click the instance, and then select **Attach Network Interface**.
4. In the **Attach Network Interface** dialog box, select the network interface, and then click **Yes, Attach**.

### To attach a network interface to a stopped or running instance using Network Interfaces

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Network Interfaces** in the navigation pane.
3. Select the network interface.
4. Click the **Attach** button.
5. In the **Attach Network Interface** dialog box, select the instance, and then click **Yes, Attach**.



## Detaching a Network Interface from an Instance

You can detach an secondary network interface at any time, using either the **Instances** or **Network Interfaces** pane of the Amazon EC2 console.

### To detach a network interface from an instance using Instances

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Instances** in the navigation pane.
3. Right-click the instance, and then select **Detach Network Interface**.
4. In the **Detach Network Interface** dialog box, select the network interface, and then click **Yes, Detach**.

### To detach a network interface from an instance using Network Interfaces

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Network Interfaces** in the navigation pane.
3. Select the network interface, and then click the **Detach** button.

4. In the **Detach Network Interface** dialog box, click **Yes, Detach**. If the network interface fails to detach from the instance, select **Force**, and then try again.



## Changing the Security Group of a Network Interface

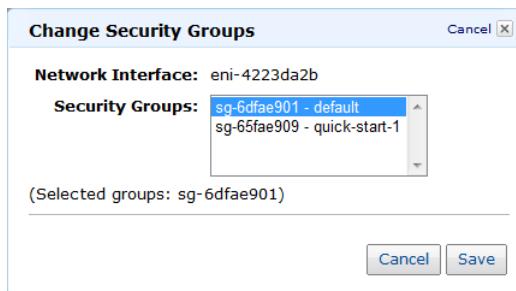
You can change the security groups that are associated with a network interface.

### Note

You can't change security group membership for interfaces owned by other Amazon Web Services, such as Elastic Load Balancing, using the Amazon EC2 console, command line interface, or API actions. To modify a security group owned by one of these services, use the console, command line interface, or API for that service.

### To change the security group of a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Network Interfaces** in the navigation pane.
3. Select the network interface.
4. Right-click the network interface, and then select **Change Security Groups**.
5. In the **Change Security Groups** dialog box, select the security groups to use, and then click **Save**.

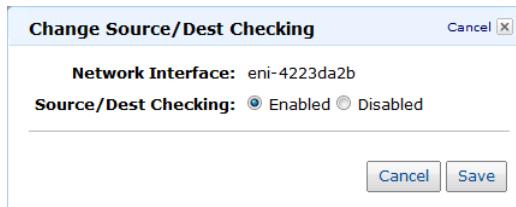


## Changing the Source/Destination Checking of a Network Interface

The Source/Destination Check attribute controls whether source/destination checking is enabled on the instance. Disabling this attribute enables an instance to handle network traffic that isn't specifically destined for the instance. For example, instances running services such as network address translation, routing, or a firewall should set this value to disabled. The default value is enabled.

### To change source/destination checking for a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Network Interfaces** in the navigation pane.
3. Right-click the network interface, and then select **Change Source/Dest Check**.
4. In the **Change Source/Dest Checking** dialog box, select **Enabled** (if enabling), or **Disabled** (if disabling), and then click **Save**.



## Associating an Elastic IP Address with a Network Interface

### To associate an Elastic IP address with a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Network Interfaces** in the navigation pane.
3. Right-click the network interface, and then select **Associate Address**.
4. In the **Associate Address** dialog box, select the Elastic IP address to associate with your network interface.
5. In **Associate to private address**, select the private IP address to associate with the Elastic IP address.
6. Click **Allow Reassociation** to allow the Elastic IP address to be associated with the specified network interface if it's currently associated with another instance or network interface, and then click **Yes, Associate**.

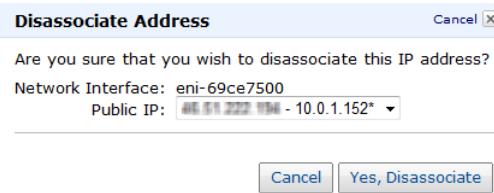


## Disassociating an Elastic IP Address from a Network Interface

If the network interface has an Elastic IP address associated with it, you can disassociate the address, and then either associate it with another network interface or release it back to the address pool.

### To disassociate an Elastic IP address from a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Network Interfaces** in the navigation pane.
3. Right-click the network interface, and then select **Disassociate Address**.
4. In the **Disassociate Address** dialog box, click **Yes, Disassociate**.



## Changing Termination Behavior for a Network Interface

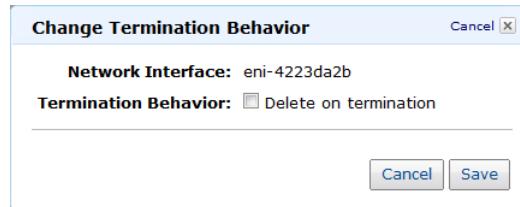
You can set the termination behavior for a network interface attached to an instance so that it is automatically deleted when you delete the instance it's attached to.

#### Note

By default, network interfaces that are automatically created and attached to instances using the EC2 console are set to terminate when the instance terminates. However, network interfaces created using the `ec2-create-network-interface` command aren't set to terminate when the instance terminates.

### To change termination behavior for network interfaces

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Network Interfaces** in the navigation pane.
3. Right-click the network interface, and then select **Change Termination Behavior**.
4. In the **Change Termination Behavior** dialog box, select the **Delete on termination** check box if you want the network interface to be deleted when you terminate an instance.



## Adding or Editing a Description for a Network Interface

### To add or edit a description for a network interface

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

2. Click **Network Interfaces** in the navigation pane.
3. Right-click the network interface, and then select **Change Description**.
4. In the **Change Description** dialog box, enter a description for the network interface, and then click **Yes, Change**.

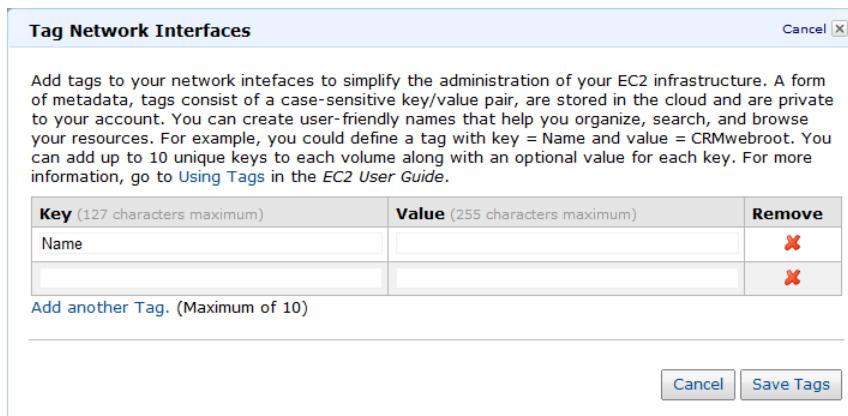


## Adding or Editing Tags for a Network Interface

Tags are metadata that you can add to a network interface. Tags are private and are only visible to your account. Each tag consists of a key and an optional value. For more information about tags, see [Tagging Your Amazon EC2 Resources \(p. 540\)](#).

### To add or edit tags for a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Network Interfaces** in the navigation pane.
3. Select the network interface.
4. In the details pane, click the **Tags** tab, and then click **Add/Edit Tags**.
5. In the **Tag Network Interfaces** dialog box, enter a key and an optional value for each tag that you want to add, and then click **Save Tags**.



## API and Command Overview

The following table summarizes the available network interface commands and corresponding API actions.

Description	Command	API Action
Attaches a network interface to an instance.	<a href="#">ec2-attach-network-interface</a>	<a href="#">AttachNetworkInterface</a>

Description	Command	API Action
Creates a network interface in the specified subnet.	<a href="#">ec2-create-network-interface</a>	CreateNetworkInterface
Deletes a network interface.	<a href="#">ec2-delete-network-interface</a>	DeleteNetworkInterface
Describes a network interface attribute.	<a href="#">ec2-describe-network-interface-attribute</a>	DescribeNetworkInterfaceAttribute
Describes one or more of your network interfaces.	<a href="#">ec2-describe-network-interfaces</a>	DescribeNetworkInterfaces
Detaches a network interface from an instance.	<a href="#">ec2-detach-network-interface</a>	DetachNetworkInterface
Modifies a network interface attribute.	<a href="#">ec2-modify-network-interface-attribute</a>	ModifyNetworkInterfaceAttribute
Resets a network interface attribute.	<a href="#">ec2-reset-network-interface-attribute</a>	ResetNetworkInterfaceAttribute

# Storage

---

## Topics

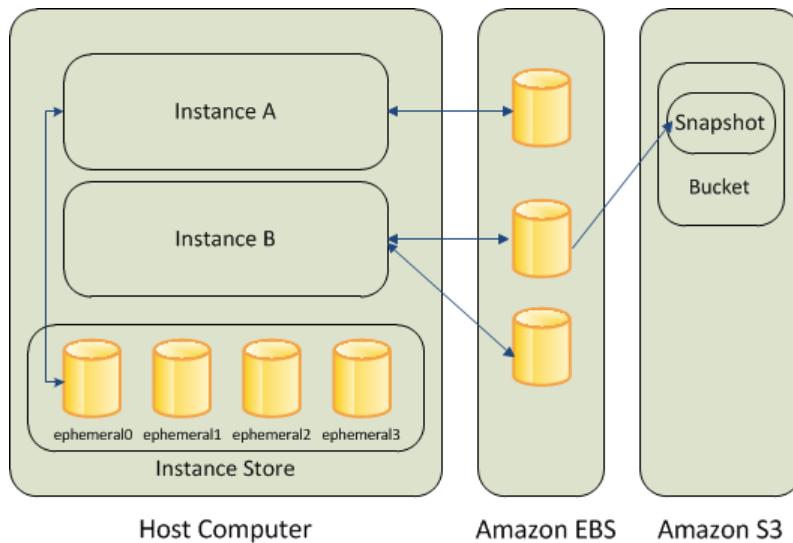
- [Amazon Elastic Block Store \(Amazon EBS\) \(p. 456\)](#)
- [Amazon EC2 Instance Store \(p. 513\)](#)
- [Amazon Simple Storage Service \(Amazon S3\) \(p. 521\)](#)
- [Block Device Mapping \(p. 522\)](#)

Amazon EC2 provides you with flexible, cost effective, and easy-to-use data storage options for your instances. Each option has a unique combination of performance and durability. These storage options can be used independently or in combination to suit your requirements.

After reading this section, you should have a good understanding about how you can use the data storage options supported by Amazon Elastic Compute Cloud to meet your specific requirements. These storage options include the following:

- Amazon Elastic Block Store (Amazon EBS)
- Amazon EC2 instance store
- Amazon Simple Storage Service (Amazon S3)

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



### Amazon EBS

Amazon EBS is a durable, block-level storage volume that you can attach to a single, running Amazon EC2 instance. You can use Amazon EBS as a primary storage device for data that requires frequent and granular updates. For example, Amazon EBS is the recommended storage option when you run a database on an instance.

Amazon EBS volumes behave like raw, unformatted, external block devices that you can attach to your instances. They persist independently from the running life of an Amazon EC2 instance. After an Amazon EBS volume is attached to an instance, you can use it like any other physical hard drive. For more information, see [Amazon Elastic Block Store \(Amazon EBS\) \(p. 456\)](#).

### Amazon EC2 Instance Store

Each Amazon EC2 instance, unless it's a micro or M3 instance, can access storage from disks that are physically attached to the host computer. This disk storage is referred to as *instance store*. Instance store provides temporary block-level storage for Amazon EC2 instances. The data on an instance store volume persists only during the life of the associated Amazon EC2 instance. For more information, see [Amazon EC2 Instance Store \(p. 513\)](#).

### Amazon S3

Amazon S3 is a repository for Internet data. Amazon S3 provides access to reliable and inexpensive data storage infrastructure. It is designed to make web-scale computing easier by enabling you to store and retrieve any amount of data, at any time, from within Amazon EC2 or anywhere on the web. For example, you can use Amazon S3 to store backup copies of your data and applications. For more information, see [Amazon Simple Storage Service \(Amazon S3\) \(p. 521\)](#).

### Adding Storage

Every time you launch an instance from an AMI, a root storage device is created for that instance. The root storage device contains all the information necessary to boot the instance. For more information, see [Amazon EC2 Root Device Volume \(p. 123\)](#). You can specify storage volumes in addition to the root device volume when you create an AMI or launch an instance using *block device mapping*. For more information, see [Block Device Mapping \(p. 522\)](#).

You can also attach EBS volumes to a running instance. For more information, see [Attaching a Volume to an Instance \(p. 469\)](#).

# Amazon Elastic Block Store (Amazon EBS)

## Topics

- [Key Features of EBS Volumes \(p. 456\)](#)
- [EBS Volume Types \(p. 458\)](#)
- [Common Tasks \(p. 459\)](#)
- [Increasing EBS Performance \(p. 461\)](#)
- [Creating or Restoring an Amazon EBS Volume \(p. 467\)](#)
- [Attaching a Volume to an Instance \(p. 469\)](#)
- [Describing Volumes \(p. 473\)](#)
- [Making an Amazon EBS Volume Available for Use \(p. 475\)](#)
- [Monitoring the Status of Your Volumes \(p. 478\)](#)
- [Detaching an Amazon EBS Volume from an Instance \(p. 489\)](#)
- [Deleting an Amazon EBS Volume \(p. 491\)](#)
- [Creating an Amazon EBS Snapshot \(p. 492\)](#)
- [Copying an Amazon EBS Snapshot \(p. 494\)](#)
- [Describing Snapshots \(p. 497\)](#)
- [Modifying Snapshot Permissions \(p. 498\)](#)
- [Deleting an Amazon EBS Snapshot \(p. 501\)](#)
- [Expanding the Storage Space of a Volume \(p. 502\)](#)
- [Using Public Data Sets \(p. 509\)](#)
- [Amazon EBS API and Command Overview \(p. 511\)](#)

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with Amazon EC2 instances. Amazon EBS volumes are highly available and reliable storage volumes that can be attached to any running instance in the same Availability Zone. The Amazon EBS volumes attached to an Amazon EBS instance are exposed as storage volumes that persist independently from the life of the instance. With Amazon EBS, you only pay for what you use. For more information about Amazon EBS pricing, see the Projecting Costs section of the [Amazon Elastic Block Store page](#).

Amazon EBS is recommended when data changes frequently and requires long-term persistence. Amazon EBS is particularly well-suited for use as the primary storage for a file system, database, or for any applications that require fine granular updates and access to raw, unformatted block-level storage. Amazon EBS is particularly helpful for database-style applications that frequently encounter many random reads and writes across the data set.

You can attach multiple volumes to the same instance within the limits specified by your AWS account. Your account has a limit on the number of Amazon EBS volumes that you can use, and the total storage available to you. For more information about these limits, and how to request an increase in your limits, see [Request to Increase the Amazon EBS Volume Limit](#).

## Key Features of EBS Volumes

All Amazon EBS volumes offer the following features:

- Data availability from replication across an Availability Zone
- Data persistence independent of the life of the instance
- The ability to create snapshots and incremental backups

### Data Availability

When you create an Amazon EBS volume in an Availability Zone, it is automatically replicated within that zone to prevent data loss due to failure of any single hardware component. After you create a volume, you can attach it to any Amazon EC2 instance in the same Availability Zone. After you attach a volume, it appears as a native block device similar to a hard drive or other physical device. At that point, the instance can interact with the volume just as it would with a local drive; the instance can format the EBS volume with a file system such as ext3 (Linux) or NTFS (Windows) and install applications. You use the file system to access the stored files.

An EBS volume can be attached to only one instance at a time within the same Availability Zone. However, multiple volumes can be attached to a single instance. If you attach multiple volumes to a device that you have named, you can stripe data across the volumes for increased I/O and throughput performance.

You can get monitoring data for your Amazon EBS volumes at no additional charge (this includes data for the root device volumes for Amazon EBS-backed instances). For more information, see [Monitoring Volumes with CloudWatch \(p. 478\)](#).

### Data Persistence

An Amazon EBS volume is off-instance storage that can persist independently from the life of an instance. You continue to pay for the volume usage as long as the data persists.

By default, EBS volumes that are attached to a running instance automatically detach from the instance with their data intact when that instance is terminated. The volume can then be reattached to a new instance, enabling quick recovery. If you are using an EBS-backed instance, you can stop and restart that instance without affecting the data stored in the attached volume. The volume remains attached throughout the stop-start cycle. This enables you to process and store the data set indefinitely, only using the processing and storage resources when required. The data set persists on the volume until the volume is deleted explicitly. After a volume is deleted, it can't be attached to any instance.

By default, EBS volumes that are created and attached to an instance at launch are deleted when that instance is terminated. You can modify this behavior by changing the value of the flag `DeleteOnTermination` to `false` when you launch the instance. This modified value causes the volume to persist even after the instance is terminated, and enables you to attach the volume to another instance.

### Snapshots

Amazon EBS provides the ability to create snapshots (backups) of any Amazon EC2 volume and write a copy of the data in the volume to Amazon S3, where it is stored redundantly in multiple Availability Zones. The volume does not need to be attached to a running instance in order to take a snapshot. As you continue to write data to a volume, you can periodically create a snapshot of the volume to use as a baseline for new volumes. These snapshots can be used to create multiple new Amazon EBS volumes, expand the size of a volume, or move volumes across Availability Zones. When you create a new volume using a snapshot, it's an exact copy of the original volume. By optionally specifying a different volume size or a different Availability Zone, you can use this functionality to increase the size of an existing volume or to create duplicate volumes in new Availability Zones. The snapshots can be shared with specific AWS accounts or made public. When you create snapshots, you incur charges in Amazon S3 based on the volume's total size. For a successive snapshot of the volume, you are only charged for any additional data beyond the volume's original size.

Amazon EBS snapshots are incremental backups, meaning that only the blocks on the volume that have changed since your last snapshot will be saved. If you have a volume with 100 GiB of data, but only 5 GiB of data have changed since your last snapshot, only the 5 GiB of modified data is written to Amazon S3. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.

To help categorize and manage your volumes and snapshots, you can tag them with metadata of your choice. For more information, see [Tagging Your Amazon EC2 Resources \(p. 540\)](#).

## EBS Volume Types

There are two types of EBS volumes:

- Standard
- Provisioned IOPS (input/output operations per second)

### Standard EBS Volumes

Standard volumes offer cost effective storage that is ideal for applications with light or bursty I/O requirements.

These volumes deliver approximately 100 IOPS on average, with burst capability of up to hundreds of IOPS.

Standard volumes are well suited for use as boot volumes, as their burst capability improves instance start-up times.

### Provisioned IOPS Volumes

To maximize the performance of your I/O-intensive applications, you can use Provisioned IOPS volumes. Provisioned IOPS volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads, that are sensitive to storage performance and consistency in random access I/O throughput. You specify an IOPS rate when you create the volume, and Amazon EBS provisions that rate for the lifetime of the volume. Amazon EBS currently supports up to 4000 IOPS per volume. You can stripe multiple volumes together to deliver thousands of IOPS per instance to your application.

A Provisioned IOPS volume must be at least 10 GB in size. The ratio of IOPS provisioned to the volume size requested can be a maximum of 10. For example, a volume with 1000 IOPS must be at least 100 GB.

Provisioned IOPS volumes are designed to offer consistent high performance. They deliver within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year, if the following guidelines are met:

- The volume is attached to an EBS-Optimized instance. For more information, see [EBS-Optimized Instances \(p. 91\)](#).
- The average queue length is at least 1 per 200 IOPS provisioned.

The queue length is the number of pending I/O requests on the volume. If you set the queue length to less than 1 per 200 IOPS provisioned, then your volume will not consistently deliver the IOPS that you've provisioned. Setting the queue length too far above the recommended setting won't affect the IOPS your volume delivers, however per-request latencies will increase.

- The read and write operations have a block size of 16 KB or less. If the I/O increases above 16 KB, the IOPS delivered drop in proportion to the increase in the size of the I/O. For example, a 1000 IOPS volume can deliver 1000 16 KB writes per second, 500 32 KB writes per second, or 250 64 KB writes per second.

As with standard EBS volumes, there is up to a 50 percent reduction in IOPS when you first access the data. Performance is restored after the data is accessed once. For maximum performance consistency with new volumes, we recommend that you read or write to all the blocks on your volume before you use it for a workload.

Note that taking a snapshot can reduce the rate of IOPS you get from your volume while the snapshot is pending. To minimize the impact of snapshots on performance of a master node, create snapshots from a read replica of your data. Ideally, create these snapshots during off-peak usage.

For more information about volume performance, see [Increasing EBS Performance \(p. 461\)](#).

## Common Tasks

This section describes some of the common tasks performed when using Amazon EBS.

### Adding Multiple Copies of Your Volume to Your EC2 Instances

1	Create a snapshot of the volume attached to your running instance.
2	Create the new volumes you need using the snapshot.
3	Attach the newly created volumes to your EC2 instances.

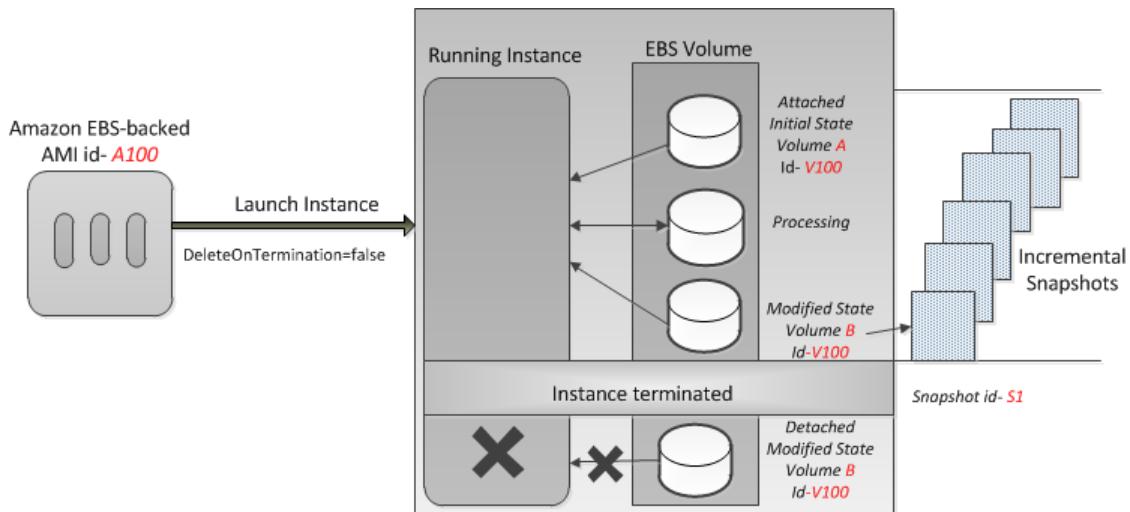
### Reducing Use of Storage Resources When Traffic Decreases

1	Identify the volume or volumes that are not currently needed.
2	[optional] Create snapshots of the identified volumes. Any AWS Marketplace product codes from the volume are propagated to the snapshot.
3	Detach the volume(s) from the instance. <b>Note</b> The volume data persists in the detached volume. You can attach a detached volume to any running instance.
4	To remove the data from the volume, delete the volume. <b>Note</b> The data in this volume is deleted. You can't attach this volume to any instance.

### Data Persistence after Instance Termination

By default, any volumes that you attach as you launch the instance are automatically deleted when the instance terminates. However, any volumes that you attached to a running instance persist even after the instance terminates. You can change the default behavior using the `DeleteOnTermination` flag. For an example of how to change this flag when launching an instance, see [Changing the Root Device Volume to Persist \(p. 128\)](#).

The following diagram and task list describe the different states of a volume from the time it is attached to an instance until the time the instance terminates.



### Persisting Data after Instance Termination

- |   |                                                                                                                                                                  |
|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Launch an instance from an EBS-backed AMI and set the value of <code>DeleteOnTermination</code> flag to <code>false</code> . The volume is in its initial state. |
| 2 | Create snapshots of the volume at regular intervals.                                                                                                             |
| 3 | Make changes to the data in the volume by doing some processing. The volume is now in a modified state.                                                          |
| 4 | Terminate the instance. The volume is detached from the instance.                                                                                                |

The volume, after instance termination, persists in its modified state. This is a new and different volume; it's no longer in its initial state or associated with its original AMI.

When you launch an instance of a particular AMI and then terminate the instance, the detached volume retains its original volume ID. If you launch another instance of the same AMI, the volume associated with the newly launched instance has a different volume ID.

### Using the Modified Volume (Volume B) after Instance Termination

- |   |                                                                                                             |
|---|-------------------------------------------------------------------------------------------------------------|
| 1 | Create a new EBS-backed AMI from the snapshot of the detached volume.                                       |
| 2 | Launch an instance using the newly created AMI. The modified (Volume B) volume is attached to the instance. |

**Or**

Attach the modified volume to another instance.

### Using the Initial Volume (Volume A) after Instance Terminates

Launch an instance from EBS-backed AMI and set the value of `DeleteOnTermination` flag to `false`.

**Note**

The instance launches with the EBS volume (Volume A) in its initial state with a new volume ID.

### Data Availability

A single Amazon EBS volume is constrained to the single Availability Zone where it is created and becomes unavailable if the Availability Zone itself is unavailable. However, a snapshot of an Amazon EBS volume is available across all of the Availability Zones within a Region. You can use these snapshots to create volumes and make them available in more than one Availability Zone.

### Making Your Data Available in Multiple Availability Zones

1	Create a snapshot of the volume that you want to make available in another Availability Zone.
2	Create a new volume using the snapshot created in the previous step, and specify a different Availability Zone.
3	Launch a new instance in that Availability Zone and attach the new volume.

## Increasing EBS Performance

After you have learned the basics of working with EBS volumes, it's a good idea to look at the I/O performance you require and your options for increasing EBS performance to meet those requirements.

### Topics

- [Maximize EBS Volume Performance \(p. 461\)](#)
- [Achieve Consistent Performance \(p. 462\)](#)
- [Monitor IOPS Performance \(p. 463\)](#)
- [Benchmark Provisioned IOPS Volumes \(p. 463\)](#)

## Maximize EBS Volume Performance

To ensure the best performance from your EBS volumes:

- Use striping for multiple volumes
- Maintain the recommended average queue length
- Read or write to all blocks on the volume before it is first used

### Striping Options for EBS Volumes

You can use RAID 0 or RAID 1+0 (RAID 10).

The following table compares these striping options.

Configuration	Use	Advantages	Disadvantages
RAID 0	When I/O performance is more important than fault tolerance, such as a heavily used database (where data replication is already set up separately).	I/O is distributed across the volumes in a stripe. If you add a volume, you get the straight addition of throughput.	Performance of the stripe is limited to the worst performing volume in the set.

Configuration	Use	Advantages	Disadvantages
RAID 1+0 (RAID 10)	When fault tolerance is more important than I/O performance, such as a critical application.	Safer from the standpoint of data durability.	Doubles the amount of I/O required from the instance to EBS compared to RAID 0, because you're mirroring all writes to a pair of volumes, limiting how much you can stripe.

### Average Queue Length

The queue length is the number of pending I/O requests. To benefit from Provisioned IOPS, a Provisioned IOPS volume must maintain an average queue length (rounded up to the nearest whole number) of 1 for every 200 provisioned IOPS in a minute. If you set the queue length to less than 1 per 200 IOPS provisioned, then your volume will not consistently deliver the IOPS that you've provisioned. Setting the queue length too far above the recommended setting won't affect the IOPS your volume delivers, however per-request latencies will increase. For a Provisioned IOPS volume of 500, the queue length average must be 3. If the average queue length is less than 3 for this volume, you aren't consistently sending enough I/O requests.

### Prepare a New Volume

There is a 5 to 50 percent reduction in IOPS when you first access the data on a volume. Performance is restored after the data is accessed once. Therefore, we recommend that you read or write to all the blocks on your volume before you use it. For a completely new volume that was created from scratch, you should write to all blocks before using the volume. For a new volume created from a snapshot, you should read all the blocks that have data before using the volume.

In Linux, the following command reads from all blocks on a volume.

```
$ dd if=/dev/md0 of=/dev/null
```

## Achieve Consistent Performance

To achieve consistent performance from your EBS volumes:

- Use EBS-Optimized instances
- Plan for snapshots

### Use EBS-Optimized Instances

The following table provides examples of when you'd want to use an EBS-Optimized instance with each type of EBS volume.

EBS Optimized instance	Volume type	Recommended use
Yes	Provisioned IOPS	Performance-sensitive production databases. Workloads with minimal variability and dedicated EBS traffic that requires high IOPS performance.
No	Provisioned IOPS	Not recommended, as Provisioned IOPS volumes are designed to deliver the expected performance only when attached to an EBS-Optimized instance.

EBS-Optimized instance	Volume type	Recommended use
Yes	Standard EBS	Scaling throughput from a number of volume stripes where I/O performance consistency isn't critical.
No	Standard EBS	Low-cost persistent storage for applications that are not performance sensitive, such as low transaction databases or log files. Also well suited for use as boot volumes.

You must use EBS-Optimized instances to get the full performance benefits of Amazon EBS Provisioned IOPS volumes. For more information, see [EBS-Optimized Instances \(p. 91\)](#).

### Plan for Snapshots

Frequent snapshots provide a higher level of data durability, but they also degrade the performance of your application while the snapshot is in progress. This trade off becomes critical when you have data that's changing rapidly.

To minimize the impact of snapshots on performance of a master node, create snapshots from a read replica of your data. Ideally, create these snapshots during off-peak usage.

## Monitor IOPS Performance

Provisioned IOPS volumes deliver within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year.

These are the key indicators of IOPS performance:

- The total throughput
- The average queue length

To ensure the consistency of Provisioned IOPS, an application must send a sufficient number of I/O requests. If an application does so, the EBS volume maintains a queue length that ensures IOPS performance. The volumes must maintain an average queue length of 1 (rounded up to the nearest whole number) for every 200 provisioned IOPS in a minute. For a Provisioned IOPS volume of 500 IOPS, the volume must maintain an average queue length of 3.

The following are possible reasons why a volume isn't meeting the performance threshold:

- The application isn't sending a sufficient number of I/O requests.
- The application is sending I/O requests that have a block size larger than 16 KB.
- [Striped volumes] The application isn't sending I/O requests randomly and consistently across all volumes in the stripe set. Every volume in the stripe set must maintain the recommended average queue length.
- Snapshots are being created frequently during peak usage.

## Benchmark Provisioned IOPS Volumes

This section demonstrates how you can test the performance of Provisioned IOPS volumes by simulating workloads similar to those of a database application. The process is as follows:

1. Launch an EBS-Optimized instance
2. Create new Provisioned IOPS volumes

3. Attach the volumes to your EBS-Optimized instance
4. Create a RAID array from the volumes, then format and mount it
5. Install a tool to benchmark I/O performance
6. Benchmark the I/O performance of your volumes
7. Delete your volumes and terminate your instance so that you don't continue to incur charges

## Set Up Your Instance

To get optimal performance from a Provisioned IOPS volume, we recommend that you use an EBS-Optimized instance. EBS-Optimized instances deliver dedicated throughput between Amazon EC2 and Amazon EBS, with options between 500 and 2000 Mbps, depending on the instance type.

To create an EBS-Optimized instance, select **Launch as an EBS-Optimized instance** when launching the instance using the EC2 console, or specify **--ebs-optimized** when using the command line. Be sure that you launch one of the instance types that supports this option. For the example tests in this topic, we recommend that you launch an `m1.xlarge` instance. For more information, see [EBS-Optimized Instances \(p. 91\)](#).

To create a Provisioned IOPS volume, select **Provisioned IOPS (io1)** when creating the volume using the EC2 console, or specify **--type io1 --iops*iops*** when using the command line. For information about attaching these volumes to your instance, see [Attaching a Volume to an Instance \(p. 469\)](#).

For the example tests, we recommend that you create a RAID array with 6 volumes, which offers a high level of performance. Because you are charged by the GB used and the number of provisioned IOPS, not the number of volumes, there is no additional cost for creating multiple, smaller volumes and using them to create a stripe set. If you're using Oracle ORION to benchmark your volumes, it can simulate striping the same way that Oracle ASM does, so we recommend that you let ORION do the striping. If you are using a different benchmarking tool, you'll need to stripe the volumes yourself.

For information about creating a striped volume on Windows, see [Create a Striped Volume in Windows](#).

To create a six-volume stripe set on Linux, use a command like the following.

```
$ sudo mdadm --create /dev/md0 --level=0 --chunk=64 --raid-devices=6 /dev/sdf
/dev/sdg /dev/sdh /dev/sdi /dev/sdj /dev/sdk
```

For this example, the file system we use is XFS. You should use the file system that meets your requirements. On Amazon Linux, use the following command to install the XFS file system.

```
$ sudo yum install -y xfsprogs
```

Then, use the following commands to create and mount the XFS file system.

```
$ sudo mkdir -p /media/p_iops_vo10 && sudo mkfs.xfs /dev/md0 &&
sudo mount -t xfs /dev/md0 /media/p_iops_vo10 && sudo chown ec2-user:ec2-user
/media/p_iops_vo10/
```

Finally, to get the most accurate results while running these tests, you must pre-warm the volume. For a completely new volume that was created from scratch, you should write to all blocks before using the volume. For a new volume created from a snapshot, you should read all the blocks that have data before using the volume. For example, on Linux you can read each block on the volume using the following command.

```
$ dd if=/dev/md0 of=/dev/null
```

On Windows, a full format of the volume pre-warms it. Use the `format <drive letter> /p:1` command to write zeros to the entire disk.

**Important**

Unless you pre-warm the volume, you might see between a 5 to 50 percent reduction in IOPS when you first access it.

## Install Benchmark Tools

The following are among the possible tools you can install and use to benchmark the performance of a Provisioned IOPS volume.

Tool	Platform	Description
<a href="#">fio</a>	Linux, Windows	For benchmarking I/O performance. (Note that fio has a dependency on libaio-devel.)
<a href="#">Calibration with the Oracle Orion Calibration Tool</a>	Linux, Windows	For calibrating the I/O performance of storage systems to be used with Oracle databases.
<a href="#">SQLIO</a>	Windows	For calibrating the I/O performance of storage systems to be used with Microsoft SQL Server.  For information about how to improve the performance of your Microsoft SQL Server databases, see <a href="#">Optimizing Databases</a> at the <i>MSDN</i> website.

## Example Benchmarking Commands

These benchmarking tools support a wide variety of test parameters. You should use commands that approximate the workloads your volumes will support. These commands are intended as examples to help you get started.

Run the following commands on an EBS-Optimized instance with attached Provisioned IOPS volumes that have been pre-warmed.

When you are finished testing your volumes, see these topics for help cleaning up: [Deleting an Amazon EBS Volume \(p. 491\)](#) and [Terminating Instances \(p. 380\)](#).

### fio Commands

Run fio on the stripe set that you created.

By default, **fio** is installed in `/usr/local/bin`, which isn't in root's path. You should update your path, or use `chmod` to enable **fio** to be run as ec2-user.

The following command performs 16 KB random write operations.

```
fio --directory=/media/p_iops_volo
--name fio_test_file --direct=1 --rw=randwrite --bs=16k --size=1G
--numjobs=16 --time_based --runtime=180 --group_reporting
```

The following command performs 16 KB random read operations.

```
fio --directory=/media/p_iops_volo
--name fio_test_file --direct=1 --rw=randread --bs=16k --size=1G
--numjobs=16 --time_based --runtime=180 --group_reporting
```

For more information about interpreting the results, go to this Linux tutorial: [Inspecting disk IO performance with fio](#).

### Oracle ORION Commands

Run ORION on the Provisioned IOPS volumes, having it simulate Oracle ASM striping instead of providing it with a stripe set that uses Windows striping.

In the directory where you installed ORION, create a file, *piops\_test.lun*, to specify the volumes for your stripe set. The following example file specifies six Provisioned IOPS volumes to be striped.

```
\\\.\D:
\\\.\E:
\\\.\F:
\\\.\G:
\\\.\H:
\\\.\I:
```

The following command performs 16 KB random I/O operations (80 percent reads and 20 percent writes), simulating 64 KB RAID-0 stripes.

```
orion -run advanced -testname piops_test -size_small 16 -size_large 16
-type rand -simulate raid0 -stripe 64 -write 80 -matrix detailed -num_disks 6
```

After the command is finished, ORION generates output files with the results in the same directory. For more information about ORION, see its [Documentation](#).

### SQLIO Commands

Run SQLIO on the stripe set that you created.

Create a file, *param.txt*, to specify your striped set. The contents of this file should look something like this (here, d:\ corresponds to the striped set, and the test uses 6 threads and a 10 GB file).

```
d:\bigtestfile.dat 6 0x0 10240
```

The following command performs 16 KB random data writes.

```
sqlio -kW -s600 -frandom -t8 -o8 -b16 -LS -BH -Fparam.txt
```

The following command performs 16 KB random data reads.

```
sqlio -kR -s600 -frandom -t8 -o8 -b16 -LS -BH -Fparam.txt
```

The results are displayed in the Command Prompt window. For more information about SQLIO, see the *readme.txt* file in your SQLIO installation directory.

# Creating or Restoring an Amazon EBS Volume

## Topics

- [AWS Management Console \(p. 467\)](#)
- [Command Line Interface \(p. 468\)](#)
- [API \(p. 469\)](#)

To use Amazon EBS, you first create a volume that can be attached to any Amazon EC2 instance within the same Availability Zone. You can create an Amazon EBS volume with data from a [snapshot stored in Amazon S3](#) or as an [empty volume](#). You can also create and attach Amazon EBS volumes when you launch instances. For more information on volumes and snapshots, see [Amazon Elastic Block Store \(Amazon EBS\) \(p. 456\)](#).

[New volumes created from existing Amazon S3 snapshots load lazily in the background](#). This means that after a volume is created from a snapshot, there is no need to wait for all of the data to transfer from Amazon S3 to your Amazon EBS volume before your attached instance can start accessing the volume and all its data. If your instance accesses data that hasn't yet been loaded, the volume immediately downloads the requested data from Amazon S3, and continues loading the rest of the data in the background.

[Accessing data for the first time from Amazon S3 might cause latency during the loading period](#). To avoid the possibility of an increased latency during the background loading of the data, you first access/read all performance-critical data from the volume (or read the entire volume) to ensure it has been loaded to the EBS volume from Amazon S3 before running the application.

## AWS Management Console

### To create or restore an Amazon EBS volume

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations \(p. 534\)](#).



3. Click **Volumes** in the navigation pane.
4. Above the upper pane, click **Create Volume**.
5. In the **Create Volume** dialog box, in the **Volume Type** drop-down list, select **Standard** or **Provisioned IOPS**. For more information, see [EBS Volume Types \(p. 458\)](#).
6. In the **Size** box and **GiB** drop-down list, select the size of the volume (in GiB or TiB).
7. For Provisioned IOPS volumes, in the **IOPS** box, enter the maximum number of input/output operations per second (IOPS) that the volume can support.
8. In the **Availability Zone** drop-down list, select the availability zone in which to launch the instance. For more information, see [Regions and Availability Zones \(p. 116\)](#)
9. To restore an Amazon EBS volume, in the **Snapshot** drop-down list, select the ID of the snapshot from which you are launching the volume. Skip this step if you aren't restoring an Amazon EBS volume.
10. Click **Yes, Create**.

**Note**

If you specify both a volume size and a snapshot ID, the size must be equal to or greater than the snapshot size. Any AWS Marketplace product codes from the snapshot are propagated to the volume.

**Note**

Any AWS Marketplace product codes from the snapshot are propagated to the volume.

## Command Line Interface

### To create or restore an Amazon EBS volume

1. To create a volume, use the [ec2-create-volume](#) command:

```
PROMPT> ec2-create-volume --size 80 --availability-zone us-east-1a
```

To restore a volume, use the [ec2-create-volume](#) command and specify the snapshot using the **--snapshot** option:

```
PROMPT> ec2-create-volume --snapshot snap-1f727354 --size 80 --availability-zone us-east-1a
```

Amazon EC2 returns information about the volume that is similar to the following example.

```
VOLUME vol-c7f95aae 80 us-east-1a creating 2010-03-30T13:54:37+0000 standard
```

2. To check whether the volume is ready, use the [ec2-describe-volumes](#) command.

```
PROMPT> ec2-describe-volumes volume_id
```

Amazon EC2 returns information about the volume that is similar to the following example.

```
VOLUME vol-c7f95aae 80 us-east-1a available 2010-03-30T13:54:37+0000 standard
```

**Note**

Any AWS Marketplace product codes from the snapshot are propagated to the volume.

## API

To create an Amazon EBS volume, use the [CreateVolume](#) action. Construct the following request.

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

The following is an example response.

```
<CreateVolumeResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
 <requestId>06eabcd-95b8-424f-87dc-da8e976f7858</requestId>
 <volumeId>vol-d11fb8b8</volumeId>
 <size>80</size>
 <snapshotId/>
 <availabilityZone>us-east-1a</availabilityZone>
 <status>creating</status>
 <createTime>2010-03-23T09:16:24.000Z</createTime>
 <volumeType>standard</volumeType>
</CreateVolumeResponse>
```

To restore an Amazon EBS volume, use the [CreateVolume](#) action. Construct the following request.

```
https://ec2.amazonaws.com/
?Action=CreateVolume
&Size=size
&SnapshotId=snapshot
&AvailabilityZone=zone
&AUTHPARAMS
```

The following is an example response.

```
<CreateVolumeResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
 <requestId>06eabcd-95b8-424f-87dc-da8e976f7858</requestId>
 <volumeId>vol-d11fb8b8</volumeId>
 <size>80</size>
 <snapshotId>snap-1f727354</snapshotId>
 <availabilityZone>us-east-1a</availabilityZone>
 <status>creating</status>
 <createTime>2010-03-23T09:16:24.000Z</createTime>
 <volumeType>standard</volumeType>
</CreateVolumeResponse>
```

## Attaching a Volume to an Instance

### Topics

- [AWS Management Console \(p. 471\)](#)
- [Command Line Interface \(p. 471\)](#)

- [API \(p. 472\)](#)

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

The following table lists the available device names on Amazon EC2. You can specify these names when attaching a volume to a running instance or when attaching a volume when launching an instance using a block device mapping. The block device driver for the instance assigns the actual volume names when mounting the volumes, and these names can be different than the names that Amazon EC2 recommends. For more information about the instance store, see [Amazon EC2 Instance Store \(p. 513\)](#). For information about the root device storage, see [Amazon EC2 Root Device Volume \(p. 123\)](#).

Instance Type	Possible for Connection	Reserved for Root	Instance Store Volumes	Recommended for Connection
Linux / UNIX	/dev/sd[a-z]  /dev/sd[a-z][1-15]  /dev/hd[a-z]  /dev/hd[a-z][1-15]	/dev/sda1	/dev/sd[b-e]	/dev/sd[f-p]  /dev/sd[f-p][1-6]
Windows	xvd[a-p]  /dev/sda[1-2]  /dev/sd[b-e]	/dev/sda1	xvd[a-e]	xvd[f-p]

**Important**

For Linux/UNIX instance types, we've received reports that some custom kernels might have restrictions that limit use to /dev/sd[f-p] or /dev/sd[f-p][1-6]. If you're having trouble using /dev/sd[q-z] or /dev/sd[q-z][1-6], try switching to /dev/sd[f-p] or /dev/sd[f-p][1-6].

After you have mapped a device using one of these device names, it is mapped to either a scsi drive (sd[a-z]) or an xvd drive (xvd[a-p]), depending on the block device driver of your kernel. If the kernel has an xvd driver, the device is mapped to xvd[a-p]. If the kernel doesn't have an xvd driver, the device is mapped to sd[a-z][1-15].

Base Windows and CentOS HPC (Cluster Compute) images come bundled with an xvd driver, so the devices on Windows and CentOS HPC (Cluster Compute) instances are mapped to xvd[a-p]. The xvd drivers (xvd[a-p]) don't support the use of trailing numbers (xvd[a-p][1-15]). We recommend that you use device names to xvd[a-p] on instances that have xvd drivers.

Depending on the size of your instance, Amazon EC2 provides instance store volumes on sd[b-e] (on Linux/UNIX) or xvd[a-e] (on Windows and CentOS HPC). Although you can connect your Amazon EBS volumes using these device names, we highly recommend that you don't because the behavior can be unpredictable.

An Amazon EC2 Windows AMI comes with an additional service installed, the **Ec2Config Service**. The Ec2Config Service runs as a local system and performs various functions to prepare an instance when it first boots up. After the devices have been mapped with the drives, the Ec2Config Service then initializes and mounts the drives. The root drive is initialized and mounted as c:\. The instance stores that comes attached to the instance are initialized and mounted as d:\, e:\, and so on. By default, when an EBS volume is attached to a Windows instance, it may show up as any drive letter on the instance. You can change the settings of the Ec2Config Service to set the drive letters of the EBS volumes per your specifications. For more information, go to [Using Ec2Config in the Amazon Elastic Compute Cloud Microsoft Windows Guide](#).

## AWS Management Console

### To attach an Amazon EBS volume to an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Volumes** in the navigation pane.

The console displays a list of current volumes.

3. Select a volume and click **Attach Volume**.

The **Attach Volume** dialog box appears.

4. Select the instance to attach the volume to from the **Instance** list box (only instances in the same Availability Zone as the volume are displayed).
5. Select how the device is exposed to the instance from the **Device** list box.
6. Click **Attach** to attach the volume to the instance.

#### Note

The volume and instance must be in the same Availability Zone.

#### Note

Windows instances use either RedHat or Citrix paravirtual drivers. Regardless of which drivers you are using, you can attach up to 16 EBS volumes using the Amazon EC2 console. If your Windows instance is using Citrix drivers, you can attach up to a total of 25 EBS volumes using the EC2 CLI. To upgrade your Windows instance from RedHat to Citrix paravirtual drivers, see [Upgrading Your PV Drivers on Your Windows AMI](#).

#### Note

If a volume has an AWS Marketplace product code:

- The volume can only be attached to the root device of a stopped instance.
- You must be subscribed to the AWS Marketplace code that is on the volume.
- The configuration (instance type, operating system) of the instance must support that specific AWS Marketplace code. For example, you cannot take a volume from a Windows instance and attach it to a Linux instance.
- AWS Marketplace product codes are copied from the volume to the instance.

For an overview of the AWS Marketplace, see <https://aws.amazon.com/marketplace/help/200900000>.  
For details on how to use the AWS Marketplace, see [AWS Marketplace](#).

## Command Line Interface

To attach an Amazon EBS volume to an instance, use [ec2-attach-volume](#).

```
PROMPT> ec2-attach-volume volume_id -i instance_id -d device
```

Amazon EC2 returns information using this syntax.

```
ATTACHMENT volume_id instance_id device attaching date_time
```

This example attaches volume `vol-4d826724` to instance `i-6058a509` in Linux and UNIX and exposes it as device `/dev/sdh`.

```
PROMPT> ec2-attach-volume vol-4d826724 -i i-6058a509 -d /dev/sdh
```

```
ATTACHMENT vol-4d826724 i-6058a509 /dev/sdh attaching 2010-03-30T13:58:58+0000
```

This example attaches volume `vol-4d826724` to instance `i-6058a509` in Windows and exposes it as device `xvdf`.

```
PROMPT> ec2-attach-volume vol-4d826724 -i i-6058a509 -d xvdf
```

```
ATTACHMENT vol-4d826724 i-6058a509 xvdf attaching 2010-03-30T13:58:58+0000
```

**Note**

The volume and instance must be in the same Availability Zone.

**Note**

Windows instances use either RedHat or Citrix paravirtual drivers. Regardless of which drivers you are using, you can attach up to 16 EBS volumes using the Amazon EC2 console. If your Windows instance is using Citrix drivers, you can attach up to a total of 25 EBS volumes using the EC2 CLI. To upgrade your Windows instance from RedHat to Citrix paravirtual drivers, see [Upgrading Your PV Drivers on Your Windows AMI](#).

**Note**

If a volume has an AWS Marketplace product code:

- The volume can only be attached to the root device of a stopped instance.
- You must be subscribed to the AWS Marketplace code that is on the volume.
- The configuration (instance type, operating system) of the instance must support that specific AWS Marketplace code. For example, you cannot take a volume from a Windows instance and attach it to a Linux instance.
- AWS Marketplace product codes are copied from the volume to the instance.

For an overview of the AWS Marketplace, see <https://aws.amazon.com/marketplace/help/200900000>. For details on how to use the AWS Marketplace, see [AWS Marketplace](#).

## API

To attach an Amazon EBS volume to an instance, use [AttachVolume](#). Construct the following request.

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

The following is an example response.

```
<AttachVolumeResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
<requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId>
<volumeId>vol-4d826724</volumeId>
<instanceId>i-6058a509</instanceId>
```

```
<device>/dev/sdh</device>
<status>attaching</status>
<attachTime>2008-05-07T11:51:50.000Z</attachTime>
</AttachVolumeResponse>
```

**Note**

The volume and instance must be in the same Availability Zone.

**Note**

Windows instances use either RedHat or Citrix paravirtual drivers. Regardless of which drivers you are using, you can attach up to 16 EBS volumes using the Amazon EC2 console. If your Windows instance is using Citrix drivers, you can attach up to a total of 25 EBS volumes using the EC2 CLI. To upgrade your Windows instance from RedHat to Citrix paravirtual drivers, see [Upgrading Your PV Drivers on Your Windows AMI](#).

**Note**

If a volume has an AWS Marketplace product code:

- The volume can only be attached to the root device of a stopped instance.
- You must be subscribed to the AWS Marketplace code that is on the volume.
- The configuration (instance type, operating system) of the instance must support that specific AWS Marketplace code. For example, you cannot take a volume from a Windows instance and attach it to a Linux instance.
- AWS Marketplace product codes are copied from the volume to the instance.

For an overview of the AWS Marketplace, see <https://aws.amazon.com/marketplace/help/200900000>. For details on how to use the AWS Marketplace, see [AWS Marketplace](#).

## Describing Volumes

**Topics**

- [AWS Management Console \(p. 473\)](#)
- [Command Line Interface \(p. 473\)](#)
- [API \(p. 474\)](#)

You can list information about a volume, including the specific instance the volume is attached to.

## AWS Management Console

### To view information about an Amazon EBS volume

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
  2. Click **Volumes** in the navigation pane.
- The console displays a list of current volumes and the instances they are attached to.
3. To view more information about a volume, select it.

Information about the volume appears in the lower pane.

## Command Line Interface

To describe volumes and list information about the volumes that you own, use the [ec2-describe-volumes](#) command.

```
PROMPT> ec2-describe-volumes
```

Amazon EC2 returns information similar to the following.

```
VOLUME vol-4d826724 80 us-east-1a in-use 2010-03-30T13:58:58+0000
 standard
ATTACHMENT vol-4d826724 i-6058a509 /dev/sdh attached 2010-03-30T13:54:55+0000
VOLUME vol-50957039 13 us-east-1a available 2010-03-24T08:01:44+0000
 standard
VOLUME vol-6682670f 1 us-east-1a in-use 2010-03-30T08:11:01+0000
 standard
ATTACHMENT vol-6682670f i-69a54000 /dev/sdh attached 2010-03-30T09:21:14+0000
```

This information includes the volume ID, capacity, status (in-use or available), and creation time of each volume. If the volume is attached, an attachment line shows the volume ID, the instance ID to which the volume is attached, the device name exposed to the instance, its status (attaching, attached, detaching, detached), and when it attached.

**Tip**

You can use **ec2-describe-volumes** to filter the results to only the volumes that match the criteria you specify.

To describe instances and list volumes that are attached to running instances, use the [ec2-describe-instances](#) command.

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

Amazon EC2 returns information similar to the following.

```
RESERVATION r-f25e6f9a 111122223333 default
INSTANCE i-84b435de ami-b232d0db ec2-184-73-201-68.compute-
1.amazonaws.comdomU-12-31-39-00-86-35.compute-1.internal running qsg-keypair
0 m1.small 2010-03-30T08:43:48+0000 us-east-1a aki-94c527fd
ari-96c527ff monitoring-disabled 184.73.201.68 10.254.137.191 ebs
BLOCKDEVICE /dev/sda1 vol-cf13b3a6 2010-03-30T08:01:44.000Z
true
BLOCKDEVICE /dev/sdh vol-c7f95aae 2010-03-30T13:58:58.000Z
true
```

**Tip**

You can use **ec2-describe-instances** to filter the results to only the instances that match the criteria you specify.

For more information about block device mapping, see [Block Device Mapping \(p. 522\)](#).

## API

To describe volumes and list information about all volumes that you own, use the [DescribeVolumes](#) action. Construct the following request.

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

The following is an example response.

```
<DescribeVolumesResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
<volumeSet>
 <item>
 <volumeId>vol-4282672b</volumeId>
 <size>80</size>
 <status>in-use</status>
 <createTime>2008-05-07T11:51:50.000Z</createTime>
 <attachmentSet>
 <item>
 <volumeId>vol-4282672b</volumeId>
 <instanceId>i-6058a509</instanceId>
 <size>80</size>
 <snapshotId>snap-12345678</snapshotId>
 <availabilityZone>us-east-1a</availabilityZone>
 <status>attached</status>
 <attachTime>2008-05-07T12:51:50.000Z</attachTime>
 </item>
 </attachmentSet>
 </item>
 ...
</volumeSet>
```

**Tip**

You can use `DescribeVolumes` to filter the results to only the volumes that match the criteria you specify.

## Making an Amazon EBS Volume Available for Use

After you attach an Amazon EBS volume to your instance, it is exposed as a block device. You can format the volume with any file system and then mount it. After you make the Amazon EBS volume available for use, you can access it in the same ways that you access any other volume. You can also take snapshots of your Amazon EBS volume for backup purposes or to use as a baseline when you create another volume.

This topic describes how to make your Amazon EBS volume available for use.

### Making the Volume Available on Linux

#### To make an Amazon EBS volume available for use on Linux

1. Connect to your instance using SSH.
2. Some Linux distributions rename your Amazon EBS volume instead of using the name that you specified when attaching the volume; for example, `/dev/sdf` through `/dev/sdp` become `/dev/xvdf` to `/dev/xvdp`. Use the `df -h` and `fdisk -l` commands to view your mounted file systems and available disk devices to help you determine the correct device name to use.
3. (Optional) Use the following command to create the ext3 file system.

**Caution**

This step assumes that you're mounting an empty volume. If you're mounting a volume that already has data on it (for example, a public data set), don't use `mkfs` before mounting the volume (skip to the next step instead). Otherwise you'll format the volume and delete the existing data.

```
$ sudo mkfs -t ext3 device_name
```

4. Use the following command to create the directory.

```
$ sudo mkdir volume_name
```

5. Use the following command to mount the volume.

```
$ sudo mount device_name volume_name
```

6. (Optional) 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 startup.

## Make the Volume Available on Windows

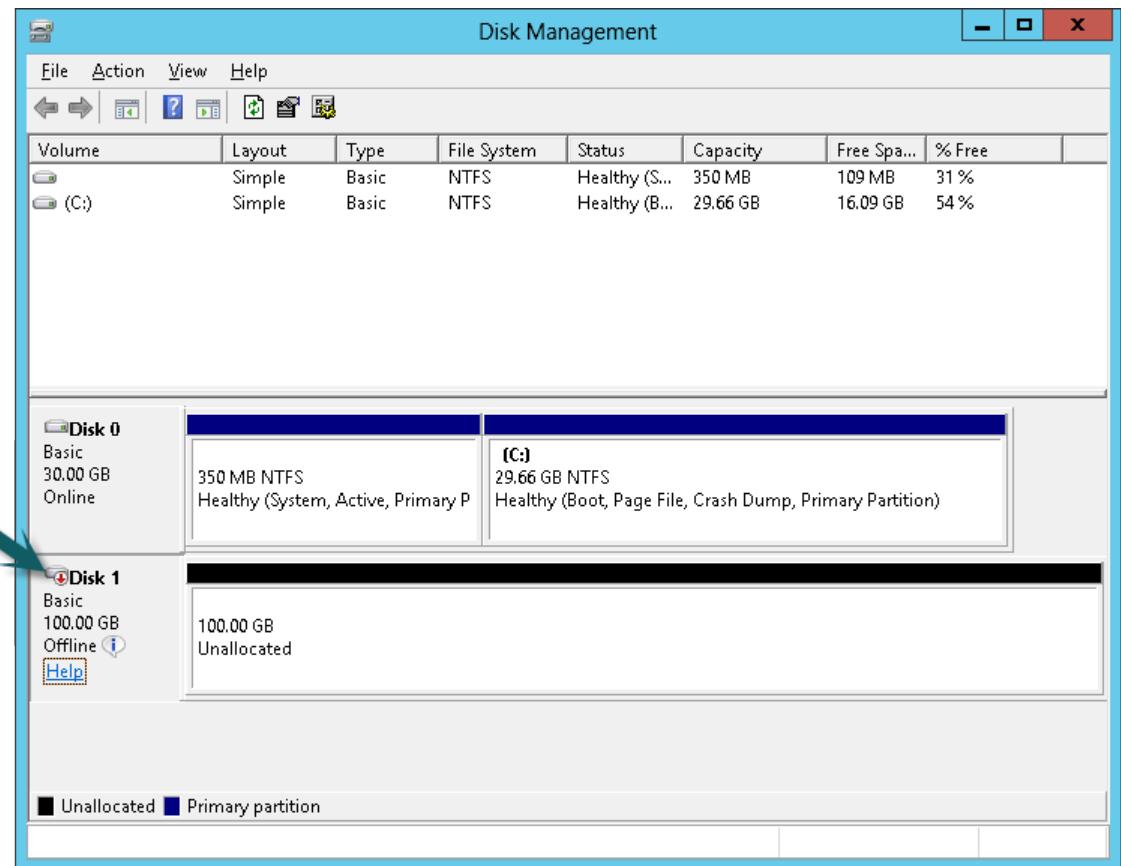
### To use an Amazon EBS volume

1. Log in to your instance using Remote Desktop.
2. • Windows Server 2008: On the taskbar, click **Start**, and then click **Run**.
  - Windows Server 2012: Go to the Start screen.
3. Type **diskmgmt.msc** and press **Enter**. The **Disk Management** utility opens.

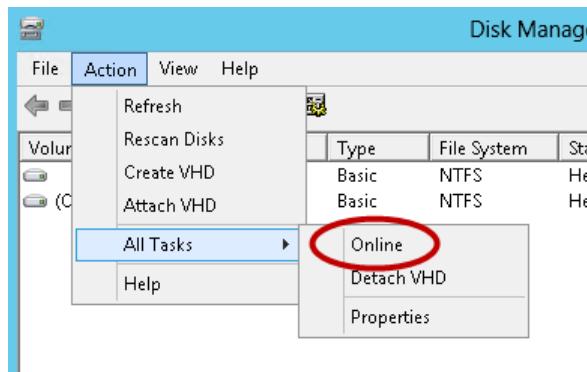
#### Caution

If you're mounting a volume that already has data on it (for example, a public data set), make sure you don't reformat the volume and delete the existing data.

4. Select the disk that represents the new Amazon EBS volume.



5. On the **Disk Management** menu, select **Action - All Tasks - Online**.



6. A new disk needs to be initialized before it can be used. To initialize the disk:
  - In the Disk Management utility, select the new Amazon EBS volume disk.
  - On the **Disk Management** menu, select **Action - All Tasks - Initialize Disk**.
  - In the **Initialize Disk** dialog, select the disk to initialize, select the desired partition style, and press **OK**.

Your new EBS volume is now available for use. Any data written to this file system is written to the Amazon EBS volume and is transparent to applications using the device.

## Monitoring the Status of Your Volumes

Amazon Web Services (AWS) automatically provides data, such as Amazon CloudWatch metrics and volume status checks, that you can use to monitor your Amazon Elastic Block Store (Amazon EBS) volumes.

### Topics

- [Monitoring Volumes with CloudWatch \(p. 478\)](#)
- [Monitoring Volumes with Status Checks \(p. 480\)](#)
- [Monitoring Volume Events \(p. 483\)](#)
- [Working with an Impaired Volume \(p. 485\)](#)
- [Working with the AutoEnableIO Volume Attribute \(p. 487\)](#)

## Monitoring Volumes with CloudWatch

CloudWatch metrics are statistical data that you can use to view, analyze, and set alarms on the operational behavior of your volumes.

The following table describes the types of monitoring data available for your Amazon EBS volumes.

Type	Description
Basic	Data is available automatically in 5-minute periods at no charge. This includes data for the root device volumes for Amazon EBS-backed instances.
Detailed	Provisioned IOPS volumes automatically send one-minute metrics to Amazon CloudWatch.

When you get data from Amazon CloudWatch, you can include a `Period` request parameter to specify the granularity of the returned data. This is different than the period that we use when we collect the data (5-minute periods). We recommend that you specify a period in your request that is equal to or larger than the collection period to ensure that the returned data is valid.

You can get the data using either the Amazon CloudWatch API or the AWS Management Console. The console takes the raw data from the Amazon CloudWatch API and displays a series of graphs based on the data. Depending on your needs, you might prefer to use either the data from the API or the graphs in the console.

## Amazon EBS Metrics

You can use the Amazon CloudWatch `GetMetricStatistics` API to get any of the Amazon EBS volume metrics listed in the following table. Similar metrics are grouped together in the table, and the metrics in the first two rows are also available for the local stores on Amazon EC2 instances.

Metric	Description
VolumeReadBytes	The total number of bytes transferred in the period. Data is only reported to Amazon CloudWatch when the volume is active. If the volume is idle, no data is reported to Amazon CloudWatch.
VolumeWriteBytes	Units: Bytes

Metric	Description
VolumeReadOps	The total number of operations in the period.
VolumeWriteOps	Units: Count
VolumeTotalReadTime	The total number of seconds spent by all operations that completed in the period. If multiple requests are submitted at the same time, this total could be greater than the length of the period. For example, say the period is 5 minutes (300 seconds); if 700 operations completed during that period, and each operation took 1 second, the value would be 700 seconds.
VolumeTotalWriteTime	Units: Seconds
VolumeIdleTime	The total number of seconds in the period when no read or write operations were submitted.  Units: Seconds
VolumeQueueLength	The number of read and write operation requests waiting to be completed in the period.  Units: Count
VolumeThroughputPercentage	Used with Provisioned IOPS volumes only. The percentage of I/O operations per second (IOPS) delivered out of the IOPS provisioned for an EBS volume. Provisioned IOPS volumes deliver within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year.  <b>Note</b> During a write, if there are no other pending I/O requests in a minute, the metric value will be 100 percent. Also, a volume's I/O performance may become degraded temporarily due to an action you have taken (e.g., creating a snapshot of a volume during peak usage, running the volume on a non-EBS-optimized instance, accessing data on the volume for the first time).  Units: Percent
VolumeConsumedReadWriteOps	Used with Provisioned IOPS volumes only. The total amount of read and write operations consumed in the period.  Units: Count

## Graphs in the AWS Management Console

After you create a volume, you can go to the Amazon EC2 console and view the volume's monitoring graphs. They're displayed when you select the volume on the **Volumes** page in the EC2 console. A **Monitoring** tab is displayed next to the volume's **Description** tab. The following table lists the graphs that are displayed. The column on the right describes how the raw data metrics from the Amazon CloudWatch API are used to produce each graph. The period for all the graphs is 5 minutes.

Graph Name	Description Using Raw Metrics
Read Bandwidth (KiB/s)	$\text{Sum}(\text{VolumeReadBytes}) / \text{Period} / 1024$
Write Bandwidth (KiB/s)	$\text{Sum}(\text{VolumeWriteBytes}) / \text{Period} / 1024$
Read Throughput (Ops/s)	$\text{Sum}(\text{VolumeReadOps}) / \text{Period}$

Graph Name	Description Using Raw Metrics
Write Throughput (Ops/s)	Sum(VolumeWriteOps) / Period
Avg Queue Length (ops)	Avg(VolumeQueueLength)
% Time Spent Idle	Sum(VolumedleTime) / Period * 100
Avg Read Size (KiB/op)	Avg(VolumeReadBytes) / 1024
Avg Write Size (KiB/op)	Avg(VolumeWriteBytes) / 1024
Avg Read Latency (ms/op)	Avg(VolumeTotalReadTime) * 1000
Avg Write Latency (ms/op)	Avg(VolumeTotalWriteTime) * 1000

For the average latency graphs and average size graphs, the average is calculated over the total number of operations (read or write, whichever is applicable to the graph) that completed during the period.

The AWS Management Console contains a console for Amazon CloudWatch. In the Amazon CloudWatch console you can search and browse all your AWS resource metrics, view graphs to troubleshoot issues and discover trends, create and edit alarms to be notified of problems, and see at-a-glance overviews of your alarms and AWS resources. For more information, see [AWS Management Console](#) in the *Amazon CloudWatch Developer Guide*.

## Monitoring Volumes with Status Checks

Volume status checks enable you to better understand, track, and manage potential inconsistencies in the data on an Amazon EBS volume. They are designed to provide you with the information that you need to determine whether your EBS volumes are impaired, and to help you control how a potentially inconsistent volume is handled.

Volume status checks are automated tests that return a pass or fail status. If all checks pass, the status of the volume is `ok`. If a check fails, the status of the volume is `impaired`. If the status is `insufficient-data`, the checks may still be in progress on the volume. You can view the results of volume status checks to identify any impaired volumes and take any necessary actions.

When EBS determines that a volume's data is potentially inconsistent, the default is that it disables I/O to the volume from any attached EC2 instances, which helps to prevent data corruption. After I/O is disabled, the next volume status check fails, and the volume status is `impaired`. In addition, you'll see an event that lets you know that I/O is disabled, and that you can resolve the impaired status of the volume by enabling I/O to the volume. We wait until you enable I/O to give you the opportunity to decide whether to continue to let your instances use the volume, or to run a consistency check using a command such as `fsck` (Linux/UNIX) or `chkdsk` (Windows) before doing so.

### Note

Volume status is based on the volume status checks, and does not reflect the volume state. Therefore, volume status does not indicate volumes in the `error` state (for example, when a volume is incapable of accepting I/O.)

If the consistency of a particular volume is not a concern for you, and you'd prefer that the volume be made available immediately if it's impaired, you can override the default behavior by configuring the volume to automatically enable I/O. If you enable the `AutoEnableIO` volume attribute, the volume status check continues to pass. In addition, you'll see an event that lets you know that the volume was determined to be potentially inconsistent, but that its I/O was automatically enabled. This enables you to check the volume's consistency or replace it at a later time.

The following table lists statuses for standard and provisioned IOPS volumes.

Overall Volume Status	I/O Enabled Status	I/O Performance Status (Provisioned IOPS volumes only)
ok	Enabled (I/O Enabled or I/O Auto-Enabled)	Normal (Volume performance is as expected)
warning	Enabled (I/O Enabled or I/O Auto-Enabled)	Degraded (Volume performance is below expectations)  Severely Degraded (Volume performance is well below expectations)
impaired	Enabled (I/O Enabled or I/O Auto-Enabled)  Disabled (Volume is offline and pending recovery, or is waiting for the user to enable I/O)	Stalled (Volume performance is severely impacted)  Not Available (Unable to determine I/O performance because I/O is disabled)
insufficient-data	Enabled (I/O Enabled or I/O Auto-Enabled)  Insufficient Data	Insufficient Data

To view and work with status checks, you can use the AWS Management Console, the API, or the command line interface.

## AWS Management Console

### To view status checks

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **ELASTIC BLOCK STORE**, click **Volumes**.
3. On the **EBS Volumes** page, the **Status Checks** column lists the operational status of each volume.
4. To view an individual volume's status, select the volume, and then click the **Status Checks** tab.

The screenshot shows the AWS Management Console interface for EBS Volumes. At the top, there's a navigation bar with 'Create Volume' and 'More...' buttons. Below it is a search bar labeled 'Viewing: All Volumes' and a 'Search' button. The main area is a table with columns: Name, Volume ID, Capacity, Volume Type, Snapshot, Created, Zone, State, Alarm Status, Attachment Information, and Monitoring. Two volumes are listed: 'empty' (Volume ID: vol-0452396b, 5 GiB, standard, created 2012-08-06T07:19:21.000Z, us-east-1b, available, none) and 'empty' (Volume ID: vol-2670f74a, 10 GiB, standard, created 2012-03-27T16:53:22.000Z, us-east-1b, deleting, none). A message at the bottom of the table says '1 Volume selected'. Below this, a detailed view for 'Volume: vol-2670f74a' is shown. It includes tabs for Details, Status Checks, Monitoring, and Tags. Under Details, it shows Volume ID (vol-2670f74a), Capacity (10 GiB), Created (2012-03-27 18:53 GMT+0200), Status (deleting), Volume Type (standard), and Product Codes (-). On the right, it shows Alarm Status (none), Snapshot, Zone (us-east-1b), Attachment, and IOPS (-). A note above the details says 'IO operations have been disabled since 104 days, 12 hours ago. Data inconsistencies may exist.' with a 'Troubleshoot Now' link.

- If you have a volume with a failed status check (status is `impaired`), see [Working with an Impaired Volume \(p. 485\)](#).

Alternatively, you can use the **Events** pane to view all events for your instances and volumes in a single pane. For more information, see [Monitoring Volume Events \(p. 483\)](#).

## Command Line Tools

The following examples show how to use the `ec2-describe-volume-status` command to describe the status of your volumes.

To do this	Run this command
Get the status of all volumes	<code>ec2-describe-volume-status</code>
Get the status of all volumes with a volume status of impaired	<code>ec2-describe-volume-status --filter "volume-status.status=impaired"</code>

The following is an example of the output for two volumes, including one that is impaired.

```
Type VolumeId AvailabilityZone VolumeStatus
VOLUME vol-0452396b us-east-1a ok
VOLUME vol-2670f74a us-east-1b impaired
Type Name Status
VOLUMESTATUS io-enabled failed
Type EventType NotBefore NotAfter EventId
 EventDescription
EVENT potential-data-inconsistency 2011-12-01T14:00:00.000Z evol-
61a54008 This is an example
Type ActionCode EventId EventType
 EventDescription
ACTION enable-volume-io evol-61a54008 potential-data-inconsistency
 This is an example
```

If you have a volume with a failed status check (status is `impaired`), see [Working with an Impaired Volume \(p. 485\)](#).

For more information, see `ec2-describe-volume-status` in the *Amazon Elastic Compute Cloud Command Reference Guide*.

## API

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

## Monitoring Volume Events

When EBS determines that a volume's data is potentially inconsistent, it disables I/O to the volume from any attached EC2 instances by default. This causes the volume status check to fail, and creates a volume status event that indicates the cause of the failure.

To automatically enable I/O on a volume with potential data inconsistencies, change the setting of the `AutoEnableIO` volume attribute. For more information about changing this attribute, see [Working with an Impaired Volume \(p. 485\)](#).

Each event includes a start time that indicates the time at which the event occurred, and a duration that indicates how long I/O for the volume was disabled. The end time is added to the event when I/O for the volume is enabled.

Volume status events include one of the following descriptions:

### Awaiting Action: Enable IO

Volume data is potentially inconsistent. I/O is disabled for the volume until you explicitly enable it. The event description changes to **IO Enabled** after you explicitly enable I/O.

### IO Enabled

I/O operations were explicitly enabled for this volume.

### IO Auto-Enabled

I/O operations were automatically enabled on this volume after an event occurred. We recommend that you check for data inconsistencies before continuing to use the data.

### Normal

For provisioned IOPS volumes only. Volume performance is as expected.

### Degraded

For provisioned IOPS volumes only. Volume performance is below expectations.

### Severely Degraded

For provisioned IOPS volumes only. Volume performance is well below expectations.

### Stalled

For provisioned IOPS volumes only. Volume performance is severely impacted.

You can view events for your volumes using the AWS Management Console, the API, or the command line interface.

## AWS Management Console

### To view events for your volumes

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Events**. You see a list of all instances and volumes and any associated events.

3. On the **Events** page, in **Viewing**, select **Volumes** to view only volume status. You can also filter on specific status types.
4. To view the event details of a specific volume, select the volume in the **Events** page.

The screenshot shows the 'Events' page in the Amazon EC2 console. The left sidebar has a 'Navigation' section with links to EC2 Dashboard, Instances, Images, Elastic Block Store, Network & Security, and more. The main area is titled 'Events' with a sub-section 'Viewing: All Resources ▾ All statuses'. A table lists seven events:

N.	Resource Id	Event Status	Resource Zone	Event Type	Description	Start Time	Duration	Event
1	vol-f138498c	IO Enabled	Volume us-east-1c	potential-data-inconsistency	IO Enabled	2012-11-07 13:49 PST	9 minutes, 18 seconds	IO
2	vol-28215055	IO Enabled	Volume us-east-1c	potential-data-inconsistency	IO Enabled	2012-11-07 15:24 PDT	6 minutes, 16 seconds	IO
3	vol-387e9b44	Awaiting Action: Enable IO	Volume us-east-1c	potential-data-inconsistency	Awaiting Action: Ena	2012-11-24 23:19 PST	2 days, 11 hours	IO
4	vol-087e9b74	IO Enabled	Volume us-east-1c	potential-data-inconsistency	IO Enabled	2012-11-14 22:35 PST	2 minutes, 7 seconds	IO
5	<input checked="" type="checkbox"/> vol-087e9b74	⚠ I/O throughput (0 to 50%) is well below exp	Volume us-east-1c	io-performance:severely-degr	I/O throughput (0 to 5	2012-11-27 11:10 PST	5 minutes, 42 seconds	IO
6	vol-c9987cb5	IO Auto Enabled	Volume us-east-1c	potential-data-inconsistency	IO Auto Enabled	2012-11-07 13:11 PST	11 minutes, 53 seconds	IO
7	vol-c9987cb5	⚠ I/O throughput (> 90%) is as expected	Volume us-east-1c	io-performance:degraded	I/O throughput (> 90%	2012-11-14 17:08 PST	1 minute	IO

Below the table, a section titled '1 Event selected' shows details for the selected event (vol-087e9b74):

- Event:** vol-087e9b74
- Event Status:** ⚠ I/O throughput (0 to 50%) is well below expectations
- IO Status:** Enabled
- Zone:** us-east-1c
- Event Type:** io-performance:severely-degraded
- Attached To:** Loading...
- Start Time:** 2012-11-27 11:10 PST
- End Time:**

For more information about Events, see [Working with Volume Status and Events in the Amazon EC2 User Guide](#). If you need technical assistance with your volume, post your issue to the [Developer Forums](#) or visit our [Support Center](#).

If you have a volume where I/O is disabled, see [Working with an Impaired Volume \(p. 485\)](#). If you have a volume where I/O performance is below normal, this might be a temporary condition due to an action you have taken (e.g., creating a snapshot of a volume during peak usage, running the volume on a non-EBS-optimized instance, accessing data on the volume for the first time, etc.).

## Command Line Tools

To view events for your volumes, use the `ec2-describe-volume-status` command as follows.

```
ec2-describe-volume-status
```

Amazon EC2 returns output similar to the following:

```
ec2-describe-volume-status vol-11111111, vol-22222222
Type VolumeId AvailabilityZone VolumeStatus
VOLUME vol-11111111 us-east-1a ok
VOLUME vol-22222222 us-east-1b impaired
Type Name Status
VOLUMESTATUS io-enabled failed
Type EventType NotBefore NotAfter EventId EventDescrip
tion
EVENT potential-data-inconsistency 2011-12-01T14:00:00.000Z 848721011
This is an example
Type ActionCode EventId EventType
EventDescription
ACTION enable-volume-io 848721011 potential-data-inconsistency
This is an example
```

For more information, see [ec2-describe-volume-status](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

If you have a volume with a failed status check, see [Working with an Impaired Volume \(p. 485\)](#).

## API

You can use the `DescribeVolumeStatus` action to retrieve the status of your volumes and any associated events. For usage information, see [DescribeVolumeStatus](#) in the *Amazon Elastic Compute Cloud API Reference*.

## Working with an Impaired Volume

This section discusses your options if a volume is impaired because the volume's data is potentially inconsistent.

### Topics

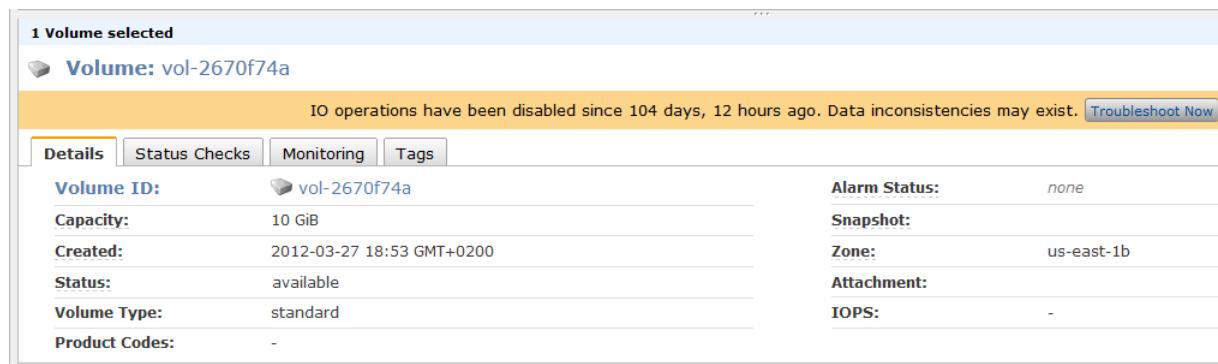
- [Option 1: Perform a Consistency Check on the Volume Attached to its Instance \(p. 485\)](#)
- [Option 2: Perform a Consistency Check on the Volume Using Another Instance \(p. 486\)](#)
- [Option 3: Delete the Volume If You No Longer Need It \(p. 487\)](#)

### Option 1: Perform a Consistency Check on the Volume Attached to its Instance

The simplest option is to enable I/O and then perform a data consistency check on the volume while the volume is still attached to its Amazon EC2 instance.

#### To perform a consistency check on an attached volume

1. Stop any applications from using the volume.
2. Enable I/O on the volume.
  - a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
  - b. In the navigation pane, under **ELASTIC BLOCK STORE**, click **Volumes**.
  - c. Select the volume on which you want to enable I/O operations.
  - d. On the **Details** tab, click **Enable Volume IO**.



- e. In **Enable Volume IO**, click **Yes, Enable**.
3. Check the data on the volume.
  - a. (Optional) Run the **fsck** (Linux) or **chkdsk** (Windows) command.
  - b. Review the application logs.

- c. (Optional) If the volume has been impaired for more than 20 minutes you can contact support. Click **Troubleshoot Now**, and then on the **Troubleshoot Status Checks** dialog box, click **Contact Support** to submit a support case.

For information about using the command line interface to enable I/O for a volume, see [ec2-enable-volume-io](#) in the *Amazon Elastic Compute Cloud Command Line Reference*. For information about using the API to enable I/O for a volume, see [EnableVolumeIO](#) in the *Amazon Elastic Compute Cloud API Reference*.

## Option 2: Perform a Consistency Check on the Volume Using Another Instance

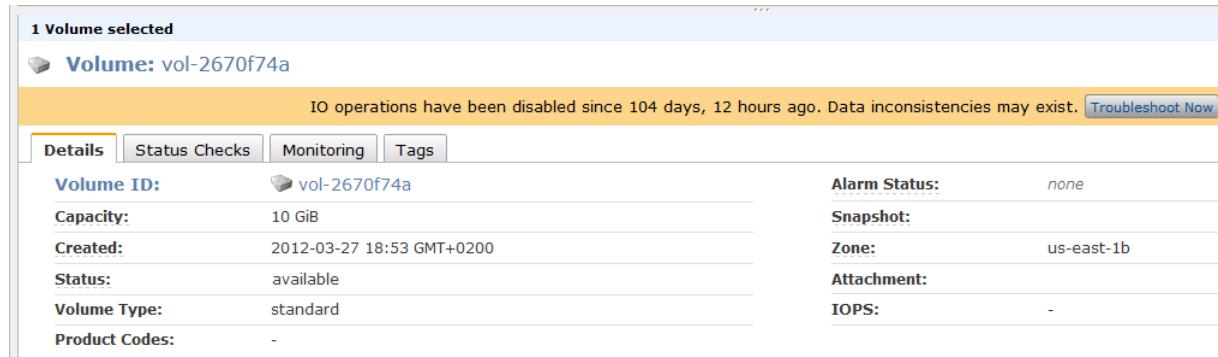
Use the following procedure to check the volume outside your production environment.

### Important

This procedure may cause the loss of write I/Os that were suspended when volume I/O was disabled.

### To perform a consistency check on a volume in isolation

1. Stop any applications from using the volume.
2. Detach the volume from the instance.
  - a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
  - b. In the navigation pane, under **ELASTIC BLOCK STORE**, click **Volumes**.
  - c. Select the volume that you want to detach.
  - d. Click **More**, and then click **Force Detach**.
  - e. In the **Force Detach Volume** dialog box, click **Yes, Force**.
3. Enable I/O on the volume.
  - a. In the navigation pane, under **ELASTIC BLOCK STORE**, click **Volumes**.
  - b. Select the volume that you detached in the previous step.
  - c. On the **Details** tab, click **Enable Volume IO**.



- d. In the **Enable Volume IO** dialog box, click **Yes, Enable**.
4. Attach the volume to another instance. For information, see [Launching EC2 Instances \(p. 274\)](#) and [Attaching a Volume to an Instance \(p. 469\)](#).

5. Check the data on the volume.
  - a. (Optional) Run the **fsck** (Linux) or **chkdsk** (Windows) command.
  - b. Review the application logs.
  - c. (Optional) If the volume has been impaired for more than 20 minutes, you can contact support. Click **Troubleshoot Now**, and then on the **Troubleshoot Status Checks** dialog box, click **Contact Support** to submit a support case.

For information about using the command line interface to enable I/O for a volume, see [ec2-enable-volume-io](#) in the *Amazon Elastic Compute Cloud Command Line Reference*. For information about using the API to enable I/O for a volume, see [EnableVolumeIO](#) in the *Amazon Elastic Compute Cloud API Reference*.

### Option 3: Delete the Volume If You No Longer Need It

If you want to remove the volume from your environment, simply delete it. For information about deleting a volume, see [Deleting an Amazon EBS Volume \(p. 491\)](#).

If you have a recent snapshot that backs up the data on the volume, you can create a new volume from the snapshot. For information about creating a volume from a snapshot, see [Creating or Restoring an Amazon EBS Volume \(p. 467\)](#).

## Working with the AutoEnableIO Volume Attribute

When EBS determines that a volume's data is potentially inconsistent, it disables I/O to the volume from any attached EC2 instances by default. This causes the volume status check to fail, and creates a volume status event that indicates the cause of the failure. If the consistency of a particular volume is not a concern, and you prefer that the volume be made available immediately if it's impaired, you can override the default behavior by configuring the volume to automatically enable I/O. If you enable the `AutoEnableIO` volume attribute, I/O between the volume and the instance is automatically reenabled and the volume's status check will pass. In addition, you'll see an event that lets you know that the volume was in a potentially inconsistent state, but that its I/O was automatically enabled. When this event occurs, you should check the volume's consistency and replace it if necessary. For more information, see [Monitoring Volume Events \(p. 483\)](#).

This section explains how to view and modify the `AutoEnableIO` attribute of a volume using the AWS Management Console, the command line interface, or the API.

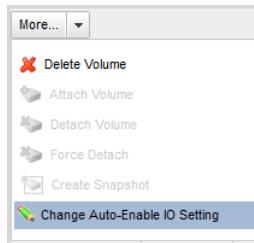
### AWS Management Console

#### To view the AutoEnableIO attribute of a volume

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **ELASTIC BLOCK STORE**, click **Volumes**.
3. Select the volume.
4. In the lower pane, click the **Status Checks** tab.
5. In the **Status Checks** tab, **Auto-Enable Volume IO** displays the current setting for your volume, either `Enabled` or `Disabled`.

### To modify the AutoEnableIO attribute of a volume

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **ELASTIC BLOCK STORE**, click **Volumes**.
3. Select the volume.
4. At the top of the **Volumes** page, click **More**.
5. Click **Change Auto-Enable IO Setting**.



6. In the **Modify Auto-Enable Volume IO** dialog box, set the **Auto-Enable Volume IO** option to automatically enable I/O for an impaired volume. To disable the feature, clear the option.

7. Click **Save**.

Alternatively, instead of completing steps 4-6 in the previous procedure, go to the **Status Checks** tab and click **Auto-Enable IO Setting**.

## Command Line Tools

Use the **ec2-describe-volume-attribute** command as follows to view the `AutoEnableIO` attribute of a volume.

```
ec2-describe-volume-attribute vol-12345678 --auto-enable-io
```

For more information, see [ec2-describe-volume-attribute](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

Use the **ec2-modify-volume-attribute** command as follows to modify the `AutoEnableIO` attribute of a volume:

```
ec2-modify-volume-attribute vol-12345678 --auto-enable-io true
```

For more information, see [ec2-modify-volume-attribute](#) in the *Amazon Elastic Compute Cloud Command Line Reference*.

## API

To view the `AutoEnableIO` attribute of a volume, use the `DescribeVolumeAttribute` action. For usage information, see [DescribeVolumeAttribute](#) in the *Amazon Elastic Compute Cloud API Reference*.

To modify the `AutoEnableIO` attribute of a volume, use the `ModifyVolumeAttribute` action. For usage information, see [ModifyVolumeAttribute](#) in the *Amazon Elastic Compute Cloud API Reference*.

# Detaching an Amazon EBS Volume from an Instance

## Topics

- [AWS Management Console \(p. 489\)](#)
- [Command Line Interface \(p. 490\)](#)
- [API \(p. 491\)](#)

You can detach an Amazon EBS volume from an instance explicitly or by terminating the instance. However, a volume must be unmounted inside the instance before being detached. Failure to do so results in the volume being stuck in the busy state while it is trying to detach, which could possibly damage the file system or the data it contains.

### Note

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

If the root volume is detached from an instance with an AWS Marketplace product code, then the AWS Marketplace product codes from that volume are no longer associated with the instance.

This example unmounts the volume and then explicitly detaches it from the instance. This is useful when you want to terminate an instance or attach a volume to a different instance. To verify that the volume is no longer attached to the instance, see [Describing Volumes \(p. 473\)](#).

## AWS Management Console

### To detach an Amazon EBS volume

1. First, unmount the volume.

For Linux/UNIX, use the following command to unmount the /dev/sdh device.

```
umount -d /dev/sdh
```

For Windows, open **Disk Management**, right-click the volume to unmount, and select **Change Drive Letter and Path**. Then, select the mount point to remove and click **Remove**.

2. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
3. Click **Volumes** in the navigation pane.

The console displays a list of current volumes.

4. Select a volume and click **Detach Volume**.
- A confirmation dialog box appears.
5. Click **Yes, Detach**.

The volume is detached from the instance.

#### Caution

If your volume stays in the *detaching* state, you can force the detachment using the **Force Detach** button. Forcing the detachment can lead to data loss or a corrupted file system. Use this option only as a last resort to detach a volume from a failed instance. The instance doesn't get an opportunity to flush file system caches or file system metadata. If you use this option, you must perform file system check and repair procedures.

## Command Line Interface

### To detach an Amazon EBS volume explicitly

1. For Linux/UNIX, use the following command to unmount the /dev/sdh device.

```
umount -d /dev/sdh
```

For Windows, open **Disk Management**, right-click the volume to unmount, and select **Change Drive Letter and Path**. Then, select the mount point to remove and click **Remove**.

2. To detach the volume, use the **ec2-detach-volume** command.

```
PROMPT> ec2-detach-volume vol-4d826724
```

Amazon EC2 returns information similar to the following example.

```
ATTACHMENT vol-4d826724 i-6058a509 /dev/sdh detaching 2010-03-30T13:58:58+0000
```

#### Caution

If your volume stays in the *detaching* state, you can force the detachment using the **--force** option of **ec2-detach-volume**. Forcing the detachment can lead to data loss or a corrupted file system. Use this option only as a last resort to detach a volume from a failed instance. The instance doesn't get an opportunity to flush file system caches or file system metadata. If you use this option, you must perform file system check and repair procedures.

## To detach an Amazon EBS volume by terminating the instance

- Use the [ec2-terminate-instances](#) command.

```
PROMPT> ec2-terminate-instances i-6058a509
```

Amazon EC2 returns information similar to the following example.

```
INSTANCE i-6058a509 running shutting-down
```

## API

To detach an Amazon EBS volume, use the [DetachVolume](#) action. Construct the following request.

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

The following is an example response.

```
<DetachVolumeResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
 <requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId>
 <volumeId>vol-4d826724</volumeId>
 <instanceId>i-6058a509</instanceId>
 <device>/dev/sdh</device>
 <status>detaching</status>
 <attachTime>2008-05-08T11:51:50.000Z</attachTime>
</DetachVolumeResponse>
```

### Caution

If your volume stays in the *detaching* state, you can force the detachment using the `Force` option of `DetachVolume`. Forcing the detachment can lead to data loss or a corrupted file system. Use this option only as a last resort to detach a volume from a failed instance. The instance will not have an opportunity to flush file system caches or file system metadata. If you use this option, you must perform file system check and repair procedures.

## Deleting an Amazon EBS Volume

### Topics

- [AWS Management Console \(p. 492\)](#)
- [Command Line Interface \(p. 492\)](#)
- [API \(p. 492\)](#)

After you no longer need a volume, you can delete it. After deletion, its data is gone and the volume can't be attached to any instance. However, before deletion, you can store a snapshot of the volume, which you can use to recreate the volume later.

This section describes how to delete a volume.

## AWS Management Console

### To delete a volume

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
  2. Click **Volumes** in the navigation pane.
- The console displays a list of current volumes.
3. Select a volume and click **Delete Volume**.
- A confirmation dialog box appears.
4. Click **Yes, Delete**.

The volume is deleted.

## Command Line Interface

To delete a volume, use the `ec2-delete-volume` command.

```
PROMPT> ec2-delete-volume vol-4282672b
```

Amazon EC2 returns information similar to the following example.

```
VOLUME vol-4282672b
```

## API

To delete a volume, use the `DeleteVolume` action. Construct the following request.

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

The following is an example response.

```
<DeleteVolumeResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
 <requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId>
 <return>true</return>
</DeleteVolumeResponse>
```

## Creating an Amazon EBS Snapshot

### Topics

- [AWS Management Console \(p. 493\)](#)
- [Command Line Interface \(p. 494\)](#)
- [API \(p. 494\)](#)

After writing data to an Amazon EBS volume, you can periodically create a snapshot of the volume to use as a baseline for new volumes or for data backup. If you make periodic snapshots of a volume, the

snapshots are incremental so that only the blocks on the device that have changed since your last snapshot are saved in the new snapshot. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.

Snapshots occur asynchronously and the status of the snapshot is pending until the snapshot is complete.

By default, only you can launch volumes from snapshots that you own. However, you can choose to share your snapshots with specific AWS accounts or make them public. For more information, see [Modifying Snapshot Permissions \(p. 498\)](#).

When a snapshot is created, any AWS Marketplace product codes from the volume are propagated to the snapshot.

You can take a snapshot of an attached volume that is in use. However, snapshots only capture data that has been written to your Amazon EBS volume at the time the snapshot command is issued. This may exclude any data that has been cached by any applications or the operating system. If you can pause any file writes to the volume long enough to take a snapshot, your snapshot should be complete. However, if you cannot pause all file writes to the volume, you need to unmount the volume from within the instance, issue the snapshot command, and then remount the volume to ensure a consistent and complete snapshot.

To create a snapshot for Amazon EBS volumes that serve as root devices, you should stop the instance before taking the snapshot.

### To unmount the volume in Linux/UNIX

- Enter the following command from the command line.

```
umount -d device_name
```

For example:

```
umount -d /dev/sdh
```

### To unmount the volume in Windows

- In Disk Management, right-click the volume to unmount, and select **Change Drive Letter and Path**.
- Select the mount point to remove and click **Remove**.

## AWS Management Console

### To create a snapshot

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- Click **Snapshots** in the navigation pane.

The console displays a list of current snapshots.

- Click **Create Snapshot**.

The Create Snapshot dialog box appears.

- Select the volume to create a snapshot for and click **Create**.

Amazon EC2 begins creating the snapshot.

## Command Line Interface

To create a snapshot, use the [ec2-create-snapshot](#) command.

```
PROMPT> ec2-create-snapshot volume_id
```

Amazon EC2 returns information similar to the following example.

```
SNAPSHOT snap-88fe11e0 vol-c7f95aae pending 2010-03-30T14:10:38+0000
111122223333 80
```

When the snapshot is complete, its status changes to completed.

## API

To create a snapshot, use the [CreateSnapshot](#) action. Construct the following request.

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

The following is an example response.

```
<CreateSnapshotResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
<requestId>fe0d60a3-b804-484e-b558-92518bebe7af</requestId>
<snapshotId>snap-05b4aa6c</snapshotId>
<volumeId>vol-d11fb88</volumeId>
<status>pending</status>
<startTime>2010-03-23T09:43:51.000Z</startTime>
<progress/>
<ownerId>999988887777</ownerId>
<volumeSize>20</volumeSize>
<description/>
</CreateSnapshotResponse>
```

## Copying an Amazon EBS Snapshot

### Topics

- [AWS Management Console \(p. 495\)](#)
- [Command Line Interface \(p. 496\)](#)
- [API \(p. 497\)](#)

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with Amazon Elastic Compute Cloud (Amazon EC2) instances. With Amazon EBS, you can create point-in-time snapshots of volumes and store them on Amazon Simple Storage Service (Amazon S3). After you've stored a snapshot in Amazon S3, you can copy it from one AWS region to another, or within the same region, using the Amazon EC2 console, Amazon EC2 CLI, or the API. You can use snapshot copies to create backups of data, to create new Amazon EBS volumes, or to create Amazon Machine Images (AMIs).

You can have up to five snapshot copy requests in progress to a single destination per account. You can copy any accessible Amazon EBS snapshots that have "completed" status, including shared snapshots and snapshots that you've created. You can also copy AWS Marketplace, VM Import/Export, and AWS Storage Gateway snapshots, but you must verify that the snapshot is supported in the destination region.

The first snapshot copy of a volume is always a full copy. Each subsequent snapshot copy is incremental, meaning that only the blocks on the volume that have changed since your last snapshot copy to the same destination are transferred. Incremental snapshots make the copy process faster. Support for incremental snapshots is specific to a region pair. For example, if you copy a snapshot from the US East (Northern Virginia) Region to the US West (Oregon) Region, the first snapshot copy of the volume will be a full copy. However, subsequent snapshot copies of the same volume transferred between the same regions will be incremental. A snapshot copy can only be done incrementally as long as there is one full copy of the same volume available in the destination region.

**Note**

You cannot copy an Amazon Relational Database Service (Amazon RDS) snapshot.

When you copy a snapshot, you are only charged for the data transfer and storage used to copy the snapshot data across regions and to store the copied snapshot in the destination region. You are not charged if the snapshot copy fails. However, if you cancel a snapshot copy that is not yet complete, or delete the source snapshot while the copy is in progress, you are charged for the bandwidth of the data transferred. Unlike the create snapshot operation, which is incremental, the copy snapshot operation copies all of the bytes in the snapshot every time. The snapshot is copied across regions using the secure Amazon Simple Storage Service Copy and the snapshot copy receives a snapshot ID that's different from the original snapshot's ID.

You can use a copy of an Amazon EBS snapshot in the following ways:

- **Geographic Expansion:** You can launch your applications in a new region.
- **Migration:** You can migrate an application to a new region, to enable better availability and minimizing cost.
- **Disaster Recovery:** You can back up your data and logs across different geographical locations at regular intervals. In case of disaster, you can restore your applications using point-in-time backups stored in the secondary region. This minimizes data loss and recovery time.

The Amazon EC2 console, Amazon EC2 CLI, and the API are designed to provide an intuitive customer experience. We use the push model in the console design to minimize user clicks for the Amazon EBS snapshot use cases discussed earlier. You can easily initiate a copy from the console by starting with the source region. We use a pull model in the API and the Amazon EC2 CLI, because these experiences factor in how customers use automation. You only need to know the source snapshot ID and source region to initiate the copy using the Amazon EC2 CLI or API.

## AWS Management Console

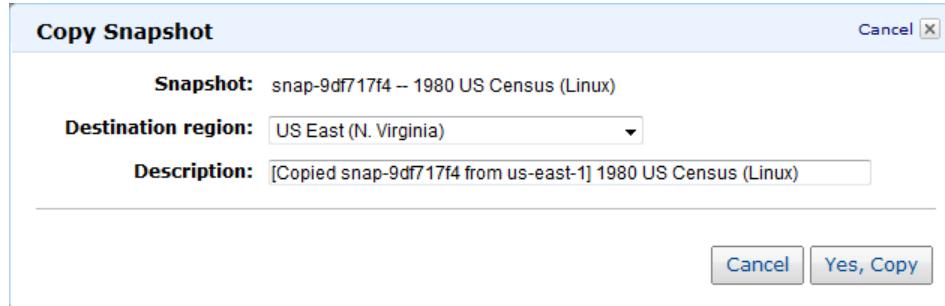
### To copy a snapshot using the Amazon EC2 console:

You can create a copy of an Amazon EBS snapshot using the Amazon EC2 console.

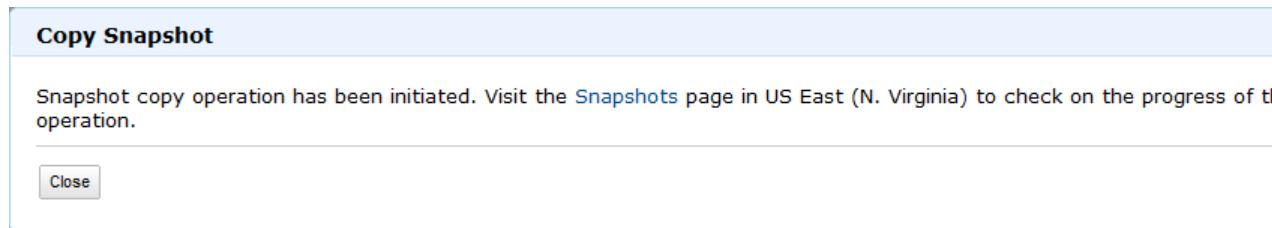
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **ELASTIC BLOCK STORE**, click **Snapshots**.
3. In the **EBS Snapshots** pane, right-click the snapshot you want to copy, and then click **Copy Snapshot**.
4. In the **Copy Snapshot** dialog box, update the following as necessary:
  - **Snapshot:** Select another snapshot from the drop-down list, if appropriate.
  - **Destination region:** Select the region where you want to write the copy of the snapshot from the drop-down list.

- **Description:** By default, the description includes information about the source snapshot so that you can identify a copy from the original. You can change this description as necessary.

5. Click **Yes, Copy**.



6. In the **Copy Snapshot** confirmation dialog box, you can click **Snapshots** to go to the **EBS Snapshots** pane in the region specified, or click **Close**.



To view the progress of the copy process later, switch the Amazon EC2 console to the destination region, and then refresh the **EBS Snapshots** pane. Copies in progress are listed at the top of the **EBS Snapshots** pane.

## Command Line Interface

You can create a copy of an Amazon EBS snapshot using the `ec2-copy-snapshot` command. You can view the progress of the copy process using the `ec2-describe-snapshots` command. For more information, see [ec2-copy-snapshot](#) and [ec2-describe-snapshots](#).

The Amazon EC2 CLI tools share a `-region` parameter that defines the region to which the API call will be directed. When copying a snapshot using the `ec2-snapshot-copy` command, you can use the `-region` parameter to specify the destination region.

### To copy a snapshot using the CLI

1. At a command prompt, switch to the destination region, and then type the following:

```
ec2-copy-snapshot -r source-region -s source-snapshot-id [-d description]
[-region destination-region]
```

2. To track the progress of the copy, at a command prompt, switch to the destination region, and then type the following:

```
ec2-describe-snapshots [snapshot_id ...] [-a] [-o owner...] [-r user_id]
[[--filter="name=value"] ...]
```

## API

### To copy a snapshot using the API

- Use the [CopySnapshot](#) action to construct the following request:

```
https://ec2.amazonaws.com/?Action=CopySnapshot
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&Description=My%20snapshot
&Signature=VjpsFPePIKxDc1IUy92W3SBApdLiap7nno4pEc9iEXAMPLE
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&SourceRegion=us-west-1
&SourceSnapshotId=snap-2EXAMPLE
&Timestamp=2012-12-11T02%3A03%3A35.713Z
&Version=2012-12-01
```

The following is an example response.

```
<CopySnapshotResponse xmlns="http://ec2.amazonaws.com/doc/2012-12-01/">
<requestId>60bc441d-fa2c-494d-b155-5d6a3EXAMPLE</requestId>
<snapshotId>snap-8EXAMPLE</snapshotId>
</CopySnapshotResponse>
```

## Describing Snapshots

### Topics

- [AWS Management Console \(p. 497\)](#)
- [Command Line Interface \(p. 498\)](#)
- [API \(p. 498\)](#)

This section describes how to view snapshots that you created.

## AWS Management Console

### To describe snapshots

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Snapshots** in the navigation pane.  
The console displays a list of all snapshots to which you have access and their status.
3. To reduce the list, select an option from the **Viewing** list box. For example, to view only your snapshots, select **Owned by Me**.  
The console displays a new list of snapshots.
4. To view more information about a snapshot, select it.

Information about the snapshot appears in the lower pane.

## Command Line Interface

To describe snapshots, use the [ec2-describe-snapshots](#) command.

```
PROMPT> ec2-describe-snapshots snap-78a54011
```

Amazon EC2 returns information about the snapshot.

```
SNAPSHOT snap-78a54011 vol-4d826724 pending 2008-02-15T09:03:58+0000 60%
```

If the snapshot is in the process of being created, the status is `pending` and a percentage complete is displayed (e.g., 60%). After the snapshot is complete, its status changes to `completed`.

**Tip**

If you have a large number of snapshots, you can use [ec2-describe-snapshots](#) to filter the results to only the snapshots that match the criteria you specify.

## API

To describe snapshots, use the [DescribeSnapshots](#) action. Construct the following request.

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

The following is an example response.

```
<DescribeSnapshotsResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
 <requestId>26245bae-2c70-4846-82d3-65784e298820</requestId>
 <snapshotSet>
 <item>
 <snapshotId>snap-05b4aa6c</snapshotId>
 <volumeId>vol-d11fbbb8</volumeId>
 <status>completed</status>
 <startTime>2010-03-23T09:43:52.000Z</startTime>
 <progress>100%</progress>
 <ownerId>999988887777</ownerId>
 <volumeSize>20</volumeSize>
 <description/>
 </item>
 </snapshotSet>
</DescribeSnapshotsResponse>
```

**Tip**

If you have a large number of snapshots, you can use [DescribeSnapshots](#) to filter the list to only the snapshots that match criteria you specify.

## Modifying Snapshot Permissions

### Topics

- [AWS Management Console \(p. 499\)](#)
- [Command Line Interface \(p. 499\)](#)
- [API \(p. 500\)](#)

This section describes how to modify permissions for your snapshots so that specific AWS accounts or all Amazon EC2 users can create volumes from them.

**Important**

When you share a snapshot (whether by sharing it with another AWS account or making it public to all), you are giving others access to all the data on your snapshot. Share snapshots only with people with whom you want to share *all* your snapshot data.

## AWS Management Console

### To modify snapshot permissions

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Snapshots** in the navigation pane.

The console displays a list of current snapshots and their status.

3. Select a snapshot and click **Permissions**.

The **Modify Snapshot Permissions** dialog box appears.

4. Choose whether to make the snapshot public or to share it with select AWS accounts:

**Important**

Making your snapshot public shares all snapshot data with everyone. Snapshots with AWS Marketplace product codes cannot be made public.

- To make the snapshot public, select **Public** and click **Save**.
- To expose the snapshot only to specific AWS accounts, select **Private**, enter the IDs of those AWS accounts, and click **Save**.

The console modifies permissions for the snapshot.

## Command Line Interface

### To modify snapshot permissions

1. Use the `ec2-describe-snapshot-attribute` command to first describe the snapshot's permissions.

```
PROMPT> ec2-describe-snapshot-attribute snap_id --create-volume-permission
```

If there are no permissions set on the snapshot, the output is empty.

2. Choose whether to make the snapshot public, or to share it with a specific AWS account.

**Important**

Making your snapshot public shares all snapshot data with everyone. Snapshots with AWS Marketplace product codes cannot be made public.

- To make the snapshot public, use the `ec2-modify-snapshot-attribute` command as follows.

```
PROMPT> ec2-modify-snapshot-attribute snap_id -c --add all
```

Amazon EC2 returns permission information for the snapshot.

```
createVolumePermission snap_id ADD group all
```

- To share the snapshot with a particular AWS account, use the [ec2-modify-snapshot-attribute](#) command as follows

```
PROMPT> ec2-modify-snapshot-attribute snap_id -c --add account_id
```

Amazon EC2 returns permission information for the snapshot.

```
createVolumePermission snap_id ADD account_id
```

## API

### To modify snapshot permissions

1. Use the [DescribeSnapshotAttribute](#) action to describe the snapshot's permissions. Construct the following request.

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

The following is an example response.

```
<DescribeSnapshotAttributeResponse xmlns="http://ec2.amazonaws.com/doc/2013-
02-01/">
 <requestId>d0d21738-e3da-4077-947d-c9e48472d831</requestId>
 <snapshotId>snap-05b4aa6c</snapshotId>
 <createVolumePermission/>
</DescribeSnapshotAttributeResponse>
```

2. Choose whether to make the snapshot public, or to share it with a specific AWS account.

**Important**

Making your snapshot public shares all snapshot data with everyone. Snapshots with AWS Marketplace product codes cannot be made public.

- To make the snapshot public, use the [ModifySnapshotAttribute](#) action. Construct the following request.

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

- To share the snapshot with a particular AWS account, use the [ModifySnapshotAttribute](#) action as follows.

```
https://ec2.amazonaws.com/
?Action=ModifySnapshotAttribute
&SnapshotId=snapshot-id
&CreateVolumePermission.Add.1.UserId=111122223333
&AUTHPARAMS
```

## Deleting an Amazon EBS Snapshot

### Topics

- [AWS Management Console \(p. 501\)](#)
- [Command Line Interface \(p. 501\)](#)
- [API \(p. 502\)](#)

This section describes how to delete a snapshot.

### Note

- If you make periodic snapshots of a volume, the snapshots are incremental so that only the blocks on the device that have changed since your last snapshot are saved in the new snapshot. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.
- You cannot delete a snapshot of the root device of an EBS volume used by a registered AMI. You must first de-register the AMI before you can delete the snapshot.

## AWS Management Console

### To delete a snapshot

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Snapshots** in the navigation pane.

The console displays a list of current snapshots.

3. Select a snapshot and click **Delete Snapshot**.  
A confirmation dialog box appears.
4. Click **Yes, Delete**.

The snapshot is deleted.

## Command Line Interface

To delete a snapshot, use the `ec2-delete-snapshot` command.

```
PROMPT> ec2-delete-snapshot snapshot_id
```

Amazon EC2 returns information similar to the following example.

```
SNAPSHOT snap-78a54011
```

## API

To delete a snapshot, use the [DeleteSnapshot](#) action. Construct the following request.

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

The following is an example response.

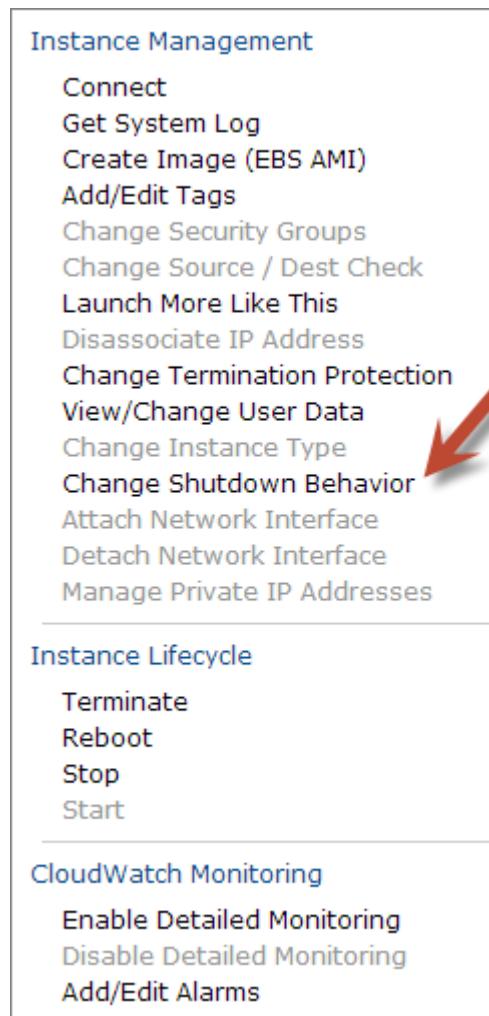
```
<DeleteSnapshotResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
 <requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId>
 <return>true</return>
</DeleteSnapshotResponse>
```

## Expanding the Storage Space of a Volume

Sometimes, it is necessary for you to increase the storage space of an existing volume without losing the data that is on the volume. This topic explains how to expand the storage space of an Amazon EBS volume, while preserving the data that is on the volume.

### To expand the storage space of an existing Amazon EBS volume

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. If the volume to be expanded is the root volume of an instance, ensure that the instance's **Shutdown Behavior** is set to **Stop** and not **Terminate**. If it is already set to **Stop**, go on to step 3.
  - a. In the navigation pane, click **Instances**, right-click on the instance to check, and select **Change Shutdown Behavior**.



- b. If the **Shutdown Behavior** is set to **Terminate**, change it to **Stop** by selecting **Stop** from the **Shutdown Behavior** list, and click **Yes, Change**.

If the **Shutdown Behavior** is set to **Stop**, click **Cancel**.



3. Stop the instance that is attached to the volume to be expanded. For more information about how to stop an instance, see [Stopping and Starting Instances \(p. 378\)](#).
4. Create a snapshot of the volume.
  - a. In the navigation pane, click **Volumes**, right-click on the volume to be expanded, and select **Create Snapshot**.
  - b. Enter a **Name** and **Description** for the snapshot, and click **Yes, Create**.
5. Create a new volume from the snapshot.
  - a. In the navigation pane, click **Snapshots**. It can take several minutes to create a snapshot.
  - b. When the status of the snapshot just created is set to **completed**, right-click the snapshot and select **Create Volume from Snapshot**.
  - c. In the **Create Volume** dialog box, select the desired **Volume Type**, enter the new **Size** for the volume, set the **Availability Zone** to the same Availability Zone as the instance, and click **Yes, Create**.
6. Detach the old volume.
  - a. In the navigation pane, click **Volumes**, select the old root volume from the list of volumes, and make note of the value of *device name* in **Attachment Information**. The **Attachment Information** value takes the following form:

*instance information:device name*
  - b. Right-click the old root volume and select **Detach Volume**.
  - c. In the **Detach Volume** dialog box, click **Yes, Detach**.
7. Attach the expanded volume
  - a. In the navigation pane, click **Volumes**, select the new root volume from the list of volumes, right-click the new volume, and select **Attach Volume**.
  - b. Select the instance from the **Instances** list, enter the same device name retrieved in step 6, and click **Yes, Attach**.
8. If the instance was stopped in step 3, restart the instance.
  - a. In the navigation pane, click **Instances**, right-click the instance, and select **Start**.
  - b. In the **Start Instances** dialog box, select **Yes, Start**. If the instance fails to start, and the volume being expanded is a root volume, verify that you attached the expanded volume using the same device name as the original volume.

After the instance has started, you need to update the instance to have it recognize the expanded volume space. This procedure varies depending on the operating system running in the instance. For information about how to do this on different operating systems, see the following topics.

- [Updating a Linux Instance \(p. 505\)](#)
- [Updating a Windows Instance \(p. 506\)](#)

After the new volume has been attached and extended in the instance, you can delete the old volume if it is no longer needed.

#### To delete the old volume

1. In the Amazon EC2 console, click **Volumes** in the navigation pane.
2. Right-click the old volume and select **Delete Volume**.
3. In the **Delete Volume** dialog box, click **Yes, Delete**.

## Updating a Linux Instance

In Linux, you use the **resize2fs** command to resize the file system to the new size of the volume. This command works even if the volume is the root volume.

#### Note

If the volume you are expanding has been partitioned, you need to increase the size of the partition before you can resize the file system. This can be accomplished with the **fdisk** or **parted** commands. Because repartitioning can inadvertently result in data loss on a volume, we highly recommend that you create a snapshot of the volume being repartitioned as a backup in case of any data loss.

1. Log in to your Linux instance using an SSH client. For more information about connecting to a Linux instance, see [Connecting to Your Linux/UNIX Instances Using SSH \(p. 295\)](#).
2. Use the **df** command to report the existing file system disk space usage. In this example, the `/dev/xvda1` device has already been expanded to 70GB, but the file system only sees the original 8GB size.

```
df -h
Filesystem Size Used Avail Use% Mounted on
/dev/xvda1 7.9G 943M 6.9G 12% /
tmpfs 1.9G 0 1.9G 0% /dev/shm
```

3. Use the **resize2fs** command to resize the file system to the new size of the volume. This command must be issued from root; otherwise, a permissions error is returned.

```
resize2fs /dev/xvda1
resize2fs 1.42.3 (14-May-2012)
Filesystem at /dev/xvda1 is mounted on /; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 5
Performing an on-line resize of /dev/xvda1 to 18350080 (4k) blocks.
The filesystem on /dev/xvda1 is now 18350080 blocks long.
```

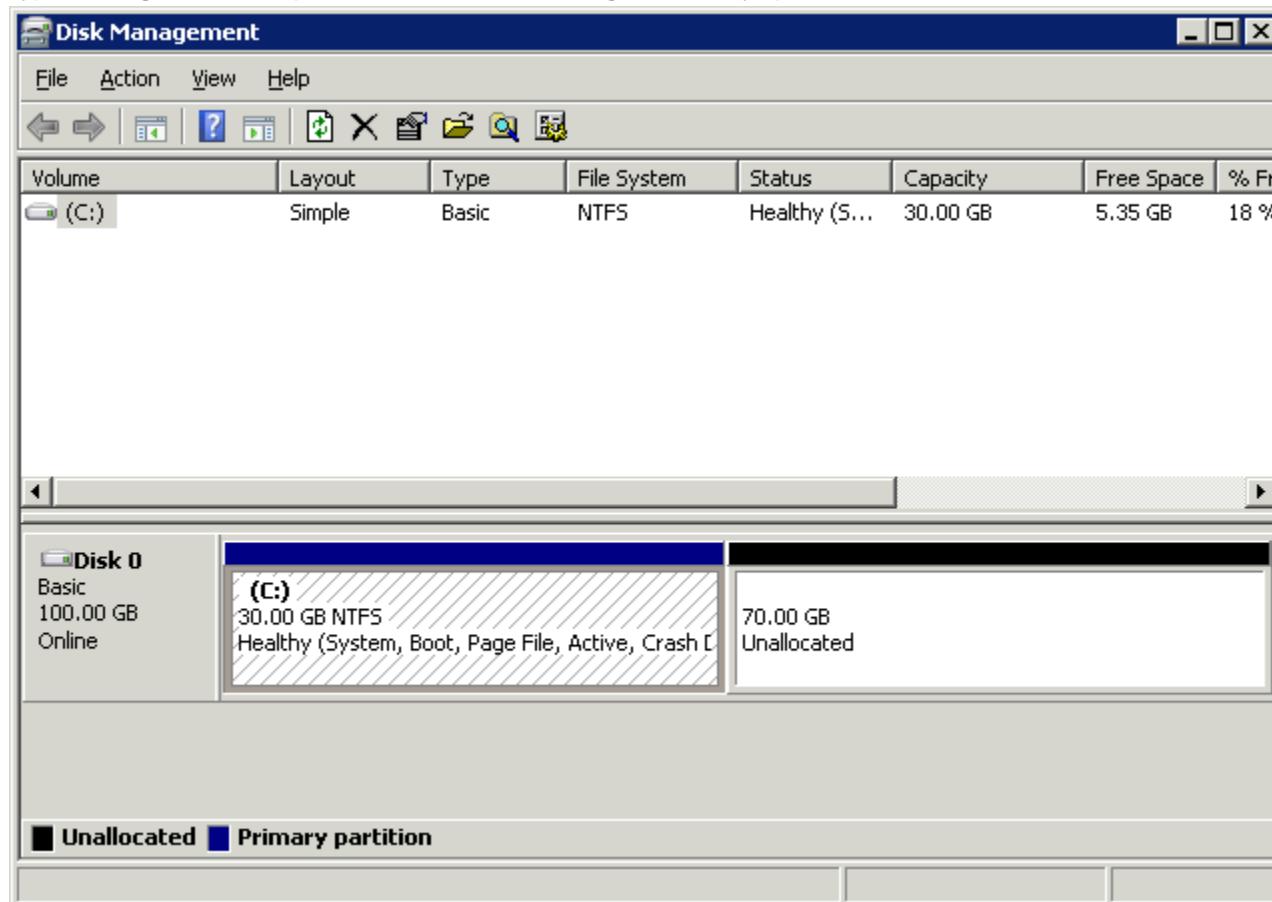
4. Use the **df** command to report the existing file system disk space usage, which should now show the full 70GB.

```
df -h
Filesystem Size Used Avail Use% Mounted on
/dev/xvda1 69G 951M 68G 2% /
tmpfs 1.9G 0 1.9G 0% /dev/shm
```

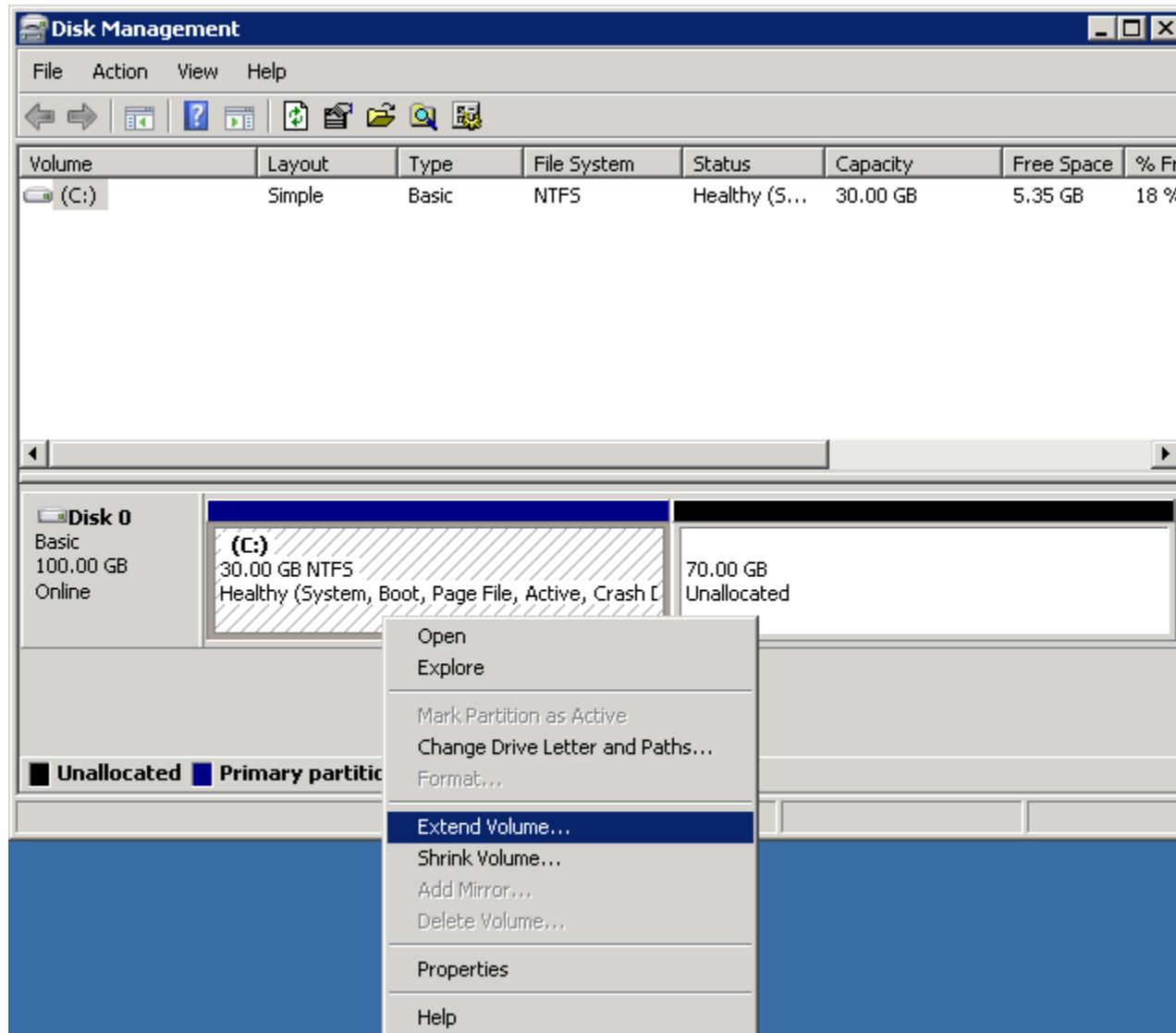
## Updating a Windows Instance

In Windows, you use the Disk Management utility to extend the disk size to the new size of the volume.

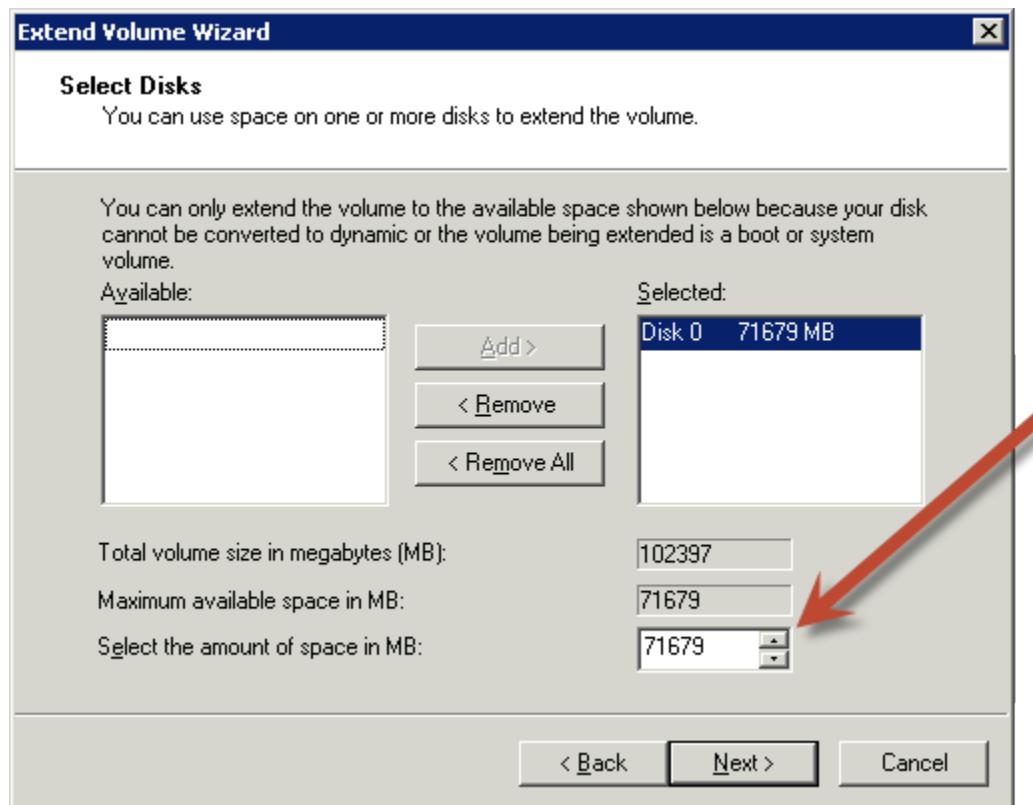
1. Log in to your Windows instance using Remote Desktop.
2. • Windows Server 2008: On the taskbar, click **Start**, and then click **Run**.  
• Windows Server 2012: Go to the Start screen.
3. Type **diskmgmt.msc** and press **Enter**. The **Disk Management** utility opens.



4. Right-click the expanded drive and select **Extend Volume**.



5. In the Extend Volume Wizard, click **Next**, then set the **Select the amount of space in MB** box to the number of megabytes by which to extend the volume. Normally, you set this to the maximum available space. Complete the wizard.



# Using Public Data Sets

## Topics

- [Public Data Set Concepts \(p. 509\)](#)
- [Finding Public Data Sets \(p. 509\)](#)
- [Launching an Instance \(p. 509\)](#)
- [Creating a Public Data Set Volume \(p. 510\)](#)
- [Mounting the Public Data Set Volume \(p. 511\)](#)

This section describes how to use Amazon EC2 public data sets.

## 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 with all AWS services, you pay only for the compute and storage you use for your 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 an Amazon EC2 instance and start computing on the data within minutes. You can also leverage the entire AWS ecosystem and easily collaborate with other AWS users. For example, you can produce or use prebuilt server images with tools and applications to analyze the data sets. By hosting this important and useful data with cost-efficient services such as Amazon EC2, AWS hopes to provide researchers across a variety of disciplines and industries with tools to enable more innovation, more quickly.

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

## Available Public Data Sets

Public data sets are currently available in the following categories:

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

## Finding Public Data Sets

Before you launch a public data set, you must locate it.

### 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 (Windows or Linux/UNIX).

## Launching an Instance

You'll attach a volume based on the public data set to an instance. Launch the instance as you typically launch an instance. For more information, see [Launching EC2 Instances \(p. 274\)](#).

## Creating a Public Data Set Volume

To use a public data set, you create an Amazon EBS volume, specifying the snapshot ID of the public data set.

### AWS Management Console

#### To create an Amazon EBS volume

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Volumes**.
3. Above the upper pane, click **Create Volume**.
4. In the **Create Volume** dialog box, in the **Volume Type** drop-down list, select **Standard** or **Provisioned IOPS**. For more information, see [EBS Volume Types \(p. 458\)](#).
5. In the **Size** box and **GiB** drop-down list, select the size of the volume (in GiB or TiB).
6. For Provisioned IOPS volumes, in the **IOPS** box, enter the maximum number of input/output operations per second (IOPS) that the volume can support.
7. In the **Availability Zone** drop-down list, select the availability zone in which to launch the instance. For more information, see [Regions and Availability Zones \(p. 116\)](#)
8. In the **Snapshot** drop-down list, select the ID of the snapshot from which you are launching the volume (optional)
9. Click **Yes, Create**.

### Command Line Tools

#### To create an Amazon EBS volume

1. Enter the following command.

```
PROMPT> ec2-create-volume --snapshot public-data-set-snapshot-id --zone
availability-zone
```

Amazon EBS returns information about the volume similar to the following example.

```
VOLUME vol-4d826724 85 us-east-1a creating 2008-02-14T00:00:00+0000
```

2. To check whether the volume is ready, use the following command.

```
PROMPT> ec2-describe-volumes vol-4d826724
```

Amazon EBS returns information about the volume similar to the following example.

```
VOLUME vol-4d826724 85 us-east-1a available 2008-07-29T08:49:25+0000
```

### API

#### To create an Amazon EBS volume

- Construct the following Query request.

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

The following is an example response.

```
<CreateVolumeResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
 <volumeId>vol-4d826724</volumeId>
 <size>85</size>
 <status>creating</status>
 <createTime>2008-05-07T11:51:50.000Z</createTime>
 <availabilityZone>us-east-1a</availabilityZone>
 <snapshotId>snap-59d33330</snapshotId>
</CreateVolumeResponse>
```

## Mounting the Public Data Set Volume

Mount the public data set volume as you typically mount an EBS volume. For more information, see [Making an Amazon EBS Volume Available for Use \(p. 475\)](#).

## Amazon EBS API and Command Overview

The following table summarizes the available Amazon EBS commands and corresponding API actions for creating and using Amazon EBS volumes.

Command and API Action	Description
<a href="#">ec2-attach-volume</a>  <a href="#">AttachVolume</a>	Attaches the specified volume to a specified instance, exposing the volume using the specified device name.  A volume can be attached to only a single instance at any time. The volume and instance must be in the same Availability Zone. The instance must be in the <code>running</code> or <code>stopped</code> state.
<a href="#">ec2-copy-snapshot</a>  <a href="#">CopySnapshot</a>	Copies a point-in-time snapshot of an Amazon Elastic Block Store (Amazon EBS) volume and stores it in Amazon Simple Storage Service (Amazon S3). You can copy the snapshot within the same region or from one region to another. You can use the snapshot to create new Amazon EBS volumes or Amazon Machine Images (AMIs).
<a href="#">ec2-create-snapshot</a>  <a href="#">CreateSnapshot</a>	Creates a snapshot of the volume you specify.  After the snapshot is created, you can use it to create volumes that contain exactly the same data as the original volume.
<a href="#">ec2-create-volume</a>  <a href="#">CreateVolume</a>	Creates a new Amazon EBS volume using the specified size and type, or based on a previously created snapshot.
<a href="#">ec2-delete-disk-image</a>	Deletes a partially or fully uploaded disk image for conversion from Amazon S3.

<b>Command and API Action</b>	<b>Description</b>
<a href="#">ec2-delete-snapshot</a>  <a href="#">DeleteSnapshot</a>	Deletes the specified snapshot.  This command does not affect currently running Amazon EBS volumes, regardless of whether they were used to create the snapshot or were derived from the snapshot.
<a href="#">ec2-delete-volume</a>  <a href="#">DeleteVolume</a>	Deletes the specified volume. The command does not delete any snapshots that were created from the volume.
<a href="#">ec2-describe-snapshot-attribute</a>  <a href="#">DescribeSnapshotAttribute</a>	Describes attributes for a snapshot.
<a href="#">ec2-describe-snapshots</a>  <a href="#">DescribeSnapshots</a>	Describes the specified snapshot.  Describes all snapshots, including their source volume, snapshot initiation time, progress (percentage complete), and status (pending, completed, and so on.).
<a href="#">ec2-describe-volumes</a>  <a href="#">DescribeVolumes</a>	Describes your volumes, including size, volume type, source snapshot, Availability Zone, creation time, status (available or in-use). If the volume is in-use, an attachment line shows the volume ID, the instance ID to which the volume is attached, the device name exposed to the instance, its status (attaching, attached, detaching, detached), and when it attached.
<a href="#">ec2-describe-volume-attribute</a>  <a href="#">DescribeVolumeAttribute</a>	Describes an attribute of a volume.
<a href="#">ec2-describe-volume-status</a>  <a href="#">DescribeVolumeStatus</a>	Describes the status of one or more volumes. Volume status provides the result of the checks performed on your volumes to determine events that can impair the performance of your volumes.
<a href="#">ec2-detach-volume</a>  <a href="#">DetachVolume</a>	Detaches the specified volume from the instance it's attached to.  This command does not delete the volume. The volume can be attached to another instance and will have the same data as when it was detached.
<a href="#">ec2-enable-volume-io</a>  <a href="#">EnableVolumeIO</a>	Enables I/O operations for a volume that had I/O operations disabled because the data on the volume was potentially inconsistent.
<a href="#">ec2-import-volume</a>  <a href="#">ImportVolume</a>	Creates a new import volume task using metadata from the specified disk image, and imports the image to Amazon EC2.
<a href="#">ec2-modify-snapshot-attribute</a>  <a href="#">ModifySnapshotAttribute</a>	Modifies permissions for a snapshot (i.e., who can create volumes from the snapshot). You can specify one or more AWS accounts, or specify all to make the snapshot public.
<a href="#">ec2-modify-volume-attribute</a>  <a href="#">ModifyVolumeAttribute</a>	Modifies a volume's attributes to determine whether a volume should be automatically enabled for I/O operations.
<a href="#">ec2-reset-snapshot-attribute</a>  <a href="#">ResetSnapshotAttribute</a>	Resets permission settings for the specified snapshot.

# Amazon EC2 Instance Store

Each Amazon EC2 instance, unless it's a micro or M3 instance, can access storage from disks that are physically attached to the host computer. This disk storage is referred to as *instance store*.

## Topics

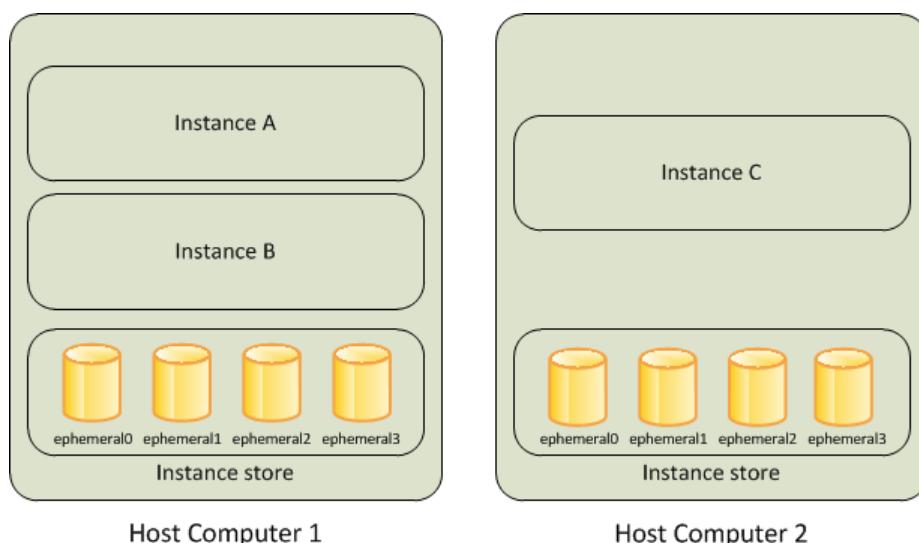
- [Instance Storage Concepts \(p. 513\)](#)
- [Instance Stores Available on Instance Types \(p. 514\)](#)
- [Instance Store Device Names \(p. 515\)](#)
- [Instance Store Usage Scenarios \(p. 516\)](#)
- [Adding Instance Store Volumes to an AMI \(p. 519\)](#)
- [Optimizing Disk Performance \(p. 520\)](#)

## Instance Storage Concepts

An Amazon EC2 instance store provides temporary block-level storage for use with an Amazon EC2 instance. The size of an instance store ranges from 150 GiB to up to 48 TiB, and varies by instance type. Larger instance types have larger instance stores. For more information, see [Instance Stores Available on Instance Types \(p. 514\)](#).

An instance store consists of one or more instance store volumes. Instance store volumes must be mounted before they can be used. By default, instances launched from an Amazon EBS-backed instance have no mounted instance store volumes. Instances launched from an instance store-backed AMI have a mounted instance store volume for the virtual machine's root device volume, and can have other mounted instance store volumes, depending on the instance type. For more information about instance store-backed AMIs and Amazon EBS-backed AMIs, see [Storage for the Root Device \(p. 19\)](#).

Instance store volumes are usable only from a single instance during its lifetime; they can't be detached and then attached to another instance. If you create an AMI from an instance, the data on its instance store volumes isn't preserved and isn't present on the instance store volumes for the instances that you launch from this AMI. While an instance store is dedicated to a particular instance, the disk subsystem is shared among instances on a host computer, as shown in the following figure.



The data in an instance store persists only during the lifetime of its associated instance. If an instance reboots (intentionally or unintentionally), data in the instance store persists. However, data on instance store volumes is lost under the following circumstances:

- Failure of an underlying drive
- Stopping an Amazon EBS-backed instance
- Terminating an instance

Therefore, do not rely on instance store volumes for valuable, long-term data. Instead, keep your data safe by using a replication strategy across multiple instances, storing data in Amazon S3, or using Amazon EBS volumes. For more information, see [Amazon Elastic Block Store \(Amazon EBS\) \(p. 456\)](#).

When you launch an instance, whether it's launched from an Amazon EBS-backed AMI or an instance store-backed AMI, you can attach instance store volumes to the instance using block device mapping. For more information, see [Adding Instance Store Volumes to an AMI \(p. 519\)](#).

## Instance Stores Available on Instance Types

Amazon EC2 instances are divided into different instance types, which determine the size of the instance store available on the instance by default. When you launch an instance, you can specify an instance type or use the default instance type, which is an `m1.small` instance.

The instance type also determines the type of hardware for your instance store volumes. A high I/O instance (`hi1.4xlarge`) uses solid state drives (SSD) to deliver very high random I/O performance. This is a good option when you need storage with very low latency, but you don't need it to persist when the instance terminates, or you can take advantage of fault tolerant architectures. For more information see [High I/O Instances \(p. 101\)](#).

The following table shows the instance types along with the size of the instance store available with each instance type.

Type	Name	Instance Store Volumes
Micro	<code>t1.micro</code>	None (use Amazon EBS volumes)
M1 Small	<code>m1.small</code>	1 x 150 GiB
M1 Medium	<code>m1.medium</code>	1 x 400 GiB
M1 Large	<code>m1.large</code>	2 x 420 GiB (840 GiB)
M1 Extra Large	<code>m1.xlarge</code>	4 x 420 GiB (1680 GiB)
M3 Extra Large	<code>m3.xlarge</code>	None (use Amazon EBS volumes)
M3 Double Extra Large	<code>m3.2xlarge</code>	None (use Amazon EBS volumes)
High-CPU Medium	<code>c1.medium</code>	1 x 340 GiB
High CPU Extra Large	<code>c1.xlarge</code>	4 x 420 GiB (1680 GiB)
High-Memory Extra Large	<code>m2.xlarge</code>	1 x 410 GiB
High-Memory Double Extra Large	<code>m2.2xlarge</code>	1 x 840 GiB
High-Memory Quadruple Extra Large	<code>m2.4xlarge</code>	2 x 840 GiB (1680 GiB)
High I/O	<code>hi1.4xlarge</code>	2 x 1 TiB SSD (2 TiB)

Type	Name	Instance Store Volumes
High Storage	hs1.8xlarge	24 x 2 TiB hard disk drives (48 TiB)
High-Memory Cluster	crl.8xlarge	2 x 120 GiB SSD (240 GiB)
Cluster Compute Quadruple Extra Large	cc1.4xlarge	2 x 840 GiB (1680 GiB)
Cluster Compute Eight Extra Large	cc2.8xlarge	4 x 840 GiB (3360 GiB)
Cluster GPU Quadruple Extra Large	cg1.4xlarge	2 x 840 GiB (1680 GiB)

## Instance Store Device Names

Within an instance store, instance store volumes are exposed as block devices. The virtual devices for instance store volumes are ephemeral[0–3]. Instance types that support one instance store volume have ephemeral0. Instance types that support two instance store volumes have ephemeral0 and ephemeral1. Instance types that support four instance store volumes have ephemeral0, ephemeral1, ephemeral2, and ephemeral3. Each instance store volume is pre-formatted with the ext3 file system. However, you can reformat volumes with the file system of your choice after you launch your instance. A Windows instance uses a built-in tool, EC2Config Service, to reformat the instance store volumes available on an instance with the NTFS file system.

Every instance store-backed AMI and instance has a mapping of the instance store volumes attached to the instance. Each entry in the mapping consists of a device name and the volume that it's mapped to. The instance store volumes are available to the instance, but you can't access them until they are mounted. A Windows instance uses the EC2Config Service to mount the instance store volumes for an instance. On Linux, the instance type determines which instance store volumes are mounted for you and which are available for you to mount yourself. The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different than the name that Amazon EC2 recommends.

The following table lists the devices names reserved for instance store volumes by instance type, and the default state of the storage device (formatted, mounted, available) on an instance store-backed instance. For example, an m1.small instance has ephemeral0 (ext3, 15 GiB) and swap (Linux swap, 896 MB).

Device Name	Linux/UNIX instance store-backed instance	Windows instance store-backed instance
/dev/sda1	Formatted and mounted as root (/).	Formatted and mounted as C:\.
/dev/sda2, xvdb	Formatted and mounted as /mnt or /media/ephemeral0 on m1.small and c1.medium instances.	Formatted and mounted on small instance types.
/dev/sda3	Formatted and mounted as /swap on m1.small and c1.medium instances.	Not available.
/dev/sdb, xvdb	Formatted and mounted as /mnt or /media/ephemeral0 on m1.medium, m1.large, m1.xlarge, c1.xlarge, cc1.4xlarge, cc2.8xlarge, crl.8xlarge, m2.xlarge, m2.2xlarge, m2.4xlarge, and hi1.4xlarge.	Formatted and mounted on m1.medium, m1.large, m1.xlarge, c1.xlarge, cc1.4xlarge, cc2.8xlarge, crl.8xlarge, m2.xlarge, m2.2xlarge, m2.4xlarge, and hi1.4xlarge.

Device Name	Linux/UNIX instance store-backed instance	Windows instance store-backed instance
/dev/sdc, xvdc	Available on m1.large, m1.xlarge, cc2.8xlarge, cc1.4xlarge, c1.xlarge, cr1.8xlarge, and hi1.4xlarge.	Formatted and mounted on m1.large, m1.xlarge, cl1.xlarge, and m2.4xlarge.
/dev/sdd, xvdd	Available on m1.xlarge, cl1.xlarge, and cc2.8xlarge.	Formatted and mounted on m1.xlarge and cl1.xlarge.
/dev/sde, xvde	Available on m1.xlarge and cl1.xlarge.	Formatted and mounted on m1.xlarge and cl1.xlarge.

An instance can have multiple instance store volumes mapped to a device. However, the number and size of these volumes must not exceed the instance store available for the instance type. For more information, see [Instance Stores Available on Instance Types \(p. 514\)](#).

## Instance Store Usage Scenarios

Instance store volumes are ideal for temporary storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers.

## Making Instance Stores Available on Your Instances

Instances that use Amazon EBS for the root device do not, by default, have instance store available at boot time. Also, you can't attach instance store volumes after you've launched an instance. Therefore, if you want your Amazon EBS-backed instance to use instance store volumes, you must specify them using a block device mapping when you create your AMI or launch your instance. Examples of block device mapping entries are: /dev/sdb=ephemeral0 and /dev/sdc=ephemeral1. For more information about block device mapping, see [Block Device Mapping \(p. 522\)](#)

The following procedure describes how to launch an Amazon EBS-backed m1.large Windows instance with instance store volumes.

### Accessing Instance Stores on Amazon EBS-backed Windows Instances

1	Locate an Amazon EBS-backed Windows AMI.
2	Add block device mapping entries.  Console: For more information, see <a href="#">To add volumes to an instance (p. 528)</a> .  CLI: Use the <code>ec2-run-instances</code> command to launch an m1.large instance; specify volumes for the block device mapping using the following options: -b "/dev/xvdb=ephemeral0" and -b "/dev/xvdc=ephemeral1"
3	Connect to the instance.
4	On the <b>Start</b> menu, choose <b>Computer</b> .
5	Devices listed: <ul style="list-style-type: none"> <li>• Local Disk C:/ 9.98GiB</li> <li>• Local Disk D:/ 419GiB</li> <li>• Local Disk E:/ 419GiB</li> </ul>

6	Double-click Local Disk C/. You can see the list of installed applications. This is your root drive.
7	Double-click Local Disk D:/ and then double-click Local Disk E:/. These drives are empty. They are the instance stores that come with your m1.large instance, and they are available for you to use with your applications.

The following procedure describes how to launch an Amazon EBS-backed m1.large Linux instance with instance store volumes.

### Accessing Instance Stores on Amazon EBS-backed Linux Instances

1	Locate an Amazon EBS-backed Linux/UNIX AMI.
2	Add block device mapping entries. Console: For more information, see <a href="#">To add volumes to an instance (p. 528)</a> . CLI: Use the <code>ec2-run-instances</code> command to launch an m1.large instance; specify volumes for the block device mapping using the following options: <code>-b "/dev/xvdb=ephemeral0"</code> and <code>-b "/dev/xvdc=ephemeral1"</code>
3	Connect to the instance.
4	Verify the instance stores currently mounted on the disk.
5	Notice a 10GiB root partition mounted on the root and 420 GiB mounted on an <code>ephemeral0</code> volume. Your m1.large instance comes with 2 420 GiB instance store volumes; the second volume is available but must be mounted before it can be used.
6	To format and mount the other 420 GiB instance store volume:  a. Create a file system of your choice on the device <code>/dev/sdc</code> (requires root privileges). b. Create a directory on which to mount the device. c. Mount the device on the newly created directory.
7	Verify that the device has been mounted.
8	Optionally, list the files on the root device.

You can also map instance store volumes to block devices when you create an AMI. The instances launched from such an AMI have instance store volumes at boot time. For information about adding a block device mapping while creating an AMI, see [Creating Your Own AMIs \(p. 30\)](#).

The following procedure describes how to access the instance store volumes from within an Amazon EC2 instance store-backed m1.large Windows instance.

### Tasks for Accessing Instance Stores on Amazon EC2 instance store-backed Windows Instances

1	Locate an Amazon EC2 instance store-backed Windows AMI.
2	Launch an m1.large instance.
3	Connect to the instance.
4	On the <b>Start</b> menu, choose <b>Computer</b> .

5	Devices listed:
	<ul style="list-style-type: none"> <li>• Local Disk C:/ 9.98GiB</li> <li>• Local Disk D:/ 419GiB</li> <li>• Local Disk E:/ 419GiB</li> </ul>
6	Double-click Local Disk C:/. You see the list of all installed applications. This is your root drive.
7	Double-click Local Disk D:/ and then double-click Local Disk E:/. These are empty. They are the instance store volumes that come with your <code>m1.large</code> instance, and they are available to use with your applications just like any physical drive.

Depending on the instance type, some instance store volumes on Amazon EC2 instance store-backed Linux and UNIX instances are not mounted when the instance is launched. For example, on an `m1.large` Linux and UNIX instance, the device `/dev/sdc`, although formatted and available, must be mounted before it can be used.

The following procedure describes how to access the instance store from within Amazon EC2 instance store-backed `m1.large` Linux instance.

#### **Accessing Instance Stores on Amazon EC2 instance store-backed Linux Instances**

1	Locate an Amazon EC2 instance store-backed Linux/Unix AMI.
2	Launch an <code>m1.large</code> instance.
3	Connect to the instance.
4	Check out the file systems currently mounted on the disk.
5	Notice 10 GiB root partition mounted on the root and 420 GiB mounted on an <code>ephemeral0</code> device. Your <code>m1.large</code> instance comes with 2 420 GiB instance store volumes; the second volume is available but must be mounted before it can be used.
6	To mount the other 420GiB: <ol style="list-style-type: none"> <li>a. Create a directory on which to mount the device.</li> <li>b. Mount the device on the newly created directory.</li> </ol>
7	Check to see whether the device has been mounted.
8	Optionally, list the files on the root device.

## **Suppressing Instance Stores at Launch Time**

You can prevent a particular instance storage volume from attaching to the instance. You can do this for both Amazon EC2 instance store-backed instances and Amazon EBS-backed instances. For example, specifying the mapping `/dev/sdc=none` when launching an instance prevents `/dev/sdc` from attaching to the instance. For more information about block device mapping, see [Block Device Mapping \(p. 522\)](#).

## Adding Instance Store Volumes to an AMI

Amazon EBS-backed AMIs don't include instance store by default. However, you might want instances launched from your Amazon EBS-backed AMIs to include instance store volumes. This section describes how to create an AMI that includes instance store volumes.

### AWS Management Console

#### To add instance store volumes to an AMI

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Instances**.
3. Select an instance and select **Create Image (EBS AMI)** from the **Actions** list.
4. In the **Create Image** dialog, click **Instance Store Volumes**.
5. For each instance store volume, select a volume from the **Instance Store** list and a device name from **Device**, and then click **Add**.

The screenshot shows the 'Create Image' dialog box. At the top, there are fields for 'Image Name\*' and 'Image Description'. A 'No Reboot' checkbox is checked. Below these are three radio buttons: 'Root Volume', 'EBS Volumes', and 'Instance Store Volumes', with 'Instance Store Volumes' selected. A section titled 'Map an instance store volume to the specified device.' contains dropdown menus for 'Instance Store: 0' and 'Device: /dev/ sdb'. An 'Add' button is located to the right of this section. Below this is a table with columns: Type, Device, Snapshot ID, Size, Volume Type, IOPS, and Delete on Termination. One row is shown for a 'Root' volume mapped to '/dev/sda1' with size 8, standard type, and true delete on termination. At the bottom, it says '0 EBS Volumes 0 Ephemerals' and 'Total size of EBS volumes: 8 GB'. A note states: 'When you create an EBS image an EBS snapshot will also be created for each of the above volumes.' At the very bottom are 'Cancel' and 'Yes, Create' buttons.

6. Click **Yes, Create**.

### Command Line Interface

Use the [ec2-register](#) command to specify a block device mapping that includes the instance store volumes for the image. For more information about block device mapping, see [Block Device Mapping \(p. 522\)](#).

#### To add instance store volumes to an AMI

1. Use the [ec2-register](#) command with the desired block device mapping information.

The following example registers an AMI with an 80GiB root device volume at /dev/sda1 created from the snap-12345678 snapshot. The root volume's DeleteOnTermination flag is set to false. The second block device mapping in the request maps /dev/sdc to ephemeral0.

You can omit the size in the block device mapping if you want to create a volume that is the size of the snapshot. If you do specify a size, it must be equal to or larger than the size of the snapshot. You can also resize the partition or create a new partition later.

**Tip**

If you're using the command line interface on a Windows computer, you must put quotation marks around any part of the command that includes an equal sign. For example: . . . -b "/dev/sdc=ephemeral0" . . .

```
PROMPT> ec2-register -n My_Image_Name -d My_image_description --root-device-name /dev/sda1 -b /dev/sda1=snap-12345678:80:false -b /dev/sdc=ephemeral0
```

In response, you get the ID for your new AMI.

IMAGE	ami-61a54008
-------	--------------

2. Any instance you launch from this AMI includes the instance store volumes that you specified when you created the AMI. You can confirm that the instance store devices are available from within the instance itself using instance metadata. For more information, see [Viewing the Instance Block Device Mapping for Instance Store Volumes \(p. 532\)](#).

## Optimizing Disk Performance

Because of the way that Amazon EC2 virtualizes disks, the first write to any location on a standard instance store volume performs more slowly than subsequent writes. (The performance of the solid state drives (SSD) used by high I/O instances is not affected this way.) For most applications, amortizing this cost over the lifetime of the instance is acceptable. However, if you require high disk performance, we recommend that you initialize your drives by writing once to every drive location before production use. If you require further improvements in latency or throughput, we recommend using Amazon EBS.

To initialize the stores, use the following Linux/UNIX dd commands, depending on which store you want to initialize (/dev/sdb, etc.).

**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).

To initialize the instance store volumes, use the following commands on the m1.large, m1.xlarge, c1.xlarge, m2.xlarge, m2.2xlarge, and m2.4xlarge instance types:

```
dd if=/dev/zero of=/dev/sdb bs=1M
dd if=/dev/zero of=/dev/sdc bs=1M
dd if=/dev/zero of=/dev/sdd bs=1M
dd if=/dev/zero of=/dev/sde bs=1M
```

For information about the instance storage that is available for each instance type, see [Instance Stores Available on Instance Types \(p. 514\)](#).

To perform initialization on all instance store volumes at the same time, use the following command:

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

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
```

## Amazon Simple Storage Service (Amazon S3)

Amazon S3 is a repository for Internet data. Amazon S3 provides access to reliable, fast, and inexpensive data storage infrastructure. It is designed to make web-scale computing easy by enabling you to store and retrieve any amount of data, at any time, from within Amazon EC2 or anywhere on the web. Amazon S3 stores data objects redundantly on multiple devices across multiple facilities and allows concurrent read or write access to these data objects by many separate clients or application threads. You can use the redundant data stored in Amazon S3 to recover quickly and reliably from instance or application failures.

Objects are the fundamental entities stored in Amazon S3. Objects consist of object data and metadata. The metadata is a set of name-value pairs that describes the object. The data portion is opaque to Amazon S3.

Every object stored in Amazon S3 is contained in a bucket. Buckets organize the Amazon S3 namespace at the highest level and identify the account responsible for that storage. Amazon S3 buckets are similar to Internet domain names. Objects stored in the buckets have a unique key value and are retrieved using a HTTP URL address. For example, if an object with a key value /photos/mygarden.jpg is stored in the myawsbucket bucket, then it is addressable using the URL <http://myawsbucket.s3.amazonaws.com/photos/mygarden.jpg>.

## Amazon S3 and Amazon EC2

The combination of Amazon S3 and Amazon EC2 provides resizable computing capacity with highly scalable and fast data storage infrastructure.

Amazon EC2 uses Amazon S3 for storing Amazon Machine Images (AMIs). You use AMIs for launching EC2 instances. In case of instance failure, you can use the stored AMI to immediately launch another instance, thereby allowing for fast recovery and business continuity.

Amazon EC2 also uses Amazon S3 to store snapshots (backup copies) of the data volumes. You can use snapshots for recovering data quickly and reliably in case of application or system failures. You can also use snapshots as a baseline to create multiple new data volumes, expand the size of an existing data volume, or move data volumes across multiple Availability Zones, thereby making your data usage highly scalable. For more information about using data volumes and snapshots, see [Amazon Elastic Block Store \(p. 456\)](#).

Amazon S3 stores AMIs and snapshots redundantly on multiple devices across multiple facilities, thus providing an instance with highly durable storage infrastructure.

Amazon EC2 doesn't have a built-in capability to work with Amazon S3. If you have permission, you can access a bucket using its URL. If you are a developer, you can use the API to access data in Amazon S3. You can also use the [AWS Tools for Windows PowerShell](#) to access Amazon S3 objects. Otherwise, there are a variety of tools that other people have written that you can use to access your data in Amazon S3; some of the common ones are discussed in the AWS forums. For more information about using Amazon S3, see the [Amazon Simple Storage Service Developer Guide](#).

# Block Device Mapping

Each Amazon EC2 instance that you launch has an associated root device volume, either an Amazon Elastic Block Store (EBS) volume or an instance store volume. You can use block device mapping to specify additional EBS volumes or instance store volumes to attach to an instance when it's launched. You can also attach additional EBS volumes to a running instance; see [Attaching a Volume to an Instance \(p. 469\)](#). However, the only way to attach instance store volumes to an instance is to use block device mapping to attach them as the instance is launched.

For more information about root device volumes, see [Changing the Root Device Volume to Persist \(p. 128\)](#).

## Topics

- [Block Device Mapping Concepts \(p. 522\)](#)
- [AMI Block Device Mapping \(p. 525\)](#)
- [Instance Block Device Mapping \(p. 528\)](#)

## Block Device Mapping Concepts

A *block device* is a storage device that moves data in sequences of bytes or bits (blocks). These devices support random access and generally use buffered I/O. Examples include hard disks, CD-ROM drives, and flash drives. A block device can be physically attached to a computer or accessed remotely as if it were physically attached to the computer. Amazon EC2 supports two types of block devices:

- instance store volumes (virtual devices whose underlying hardware is physically attached to the host computer for the instance)
- Amazon EBS volumes (remote storage devices)

A *block device mapping* defines the block devices to be attached to an Amazon EC2 instance. You can specify a block device mapping as part of creating an AMI so that the mapping is used by all instances launched from the AMI. Alternatively, you can specify a block device mapping when you launch an instance, so this mapping overrides the one specified in the AMI from which you launched the instance.

## Specifying a Block Device Mapping

Use a block device mapping to attach instance store volumes and EBS volumes to an Amazon EC2 instance.

When you create a block device mapping, you specify this information for each block device that you need to attach to the instance:

- [Linux/UNIX] The device name within Amazon EC2, as shown in this table. The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different than the name that Amazon EC2 recommends.

<b>Reserved for the root device</b>	/dev/sda1
<b>Recommended for instance store volumes</b>	/dev/sd[b-e]
<b>Recommended for EBS volumes</b>	/dev/sd[f-p] /dev/sd[f-p][1-6]

<b>Possible for EBS volumes</b>	/dev/sd[a-z] /dev/sd[a-z][1-15] (see Tip) /dev/hd[a-z] /dev/hd[a-z][1-15]
---------------------------------	------------------------------------------------------------------------------------

**Tip**

On some devices, the names that you specify in your block device mapping can conflict with the default block device names. To avoid this issue, do not use names of the form `/dev/sda[2-15]`.

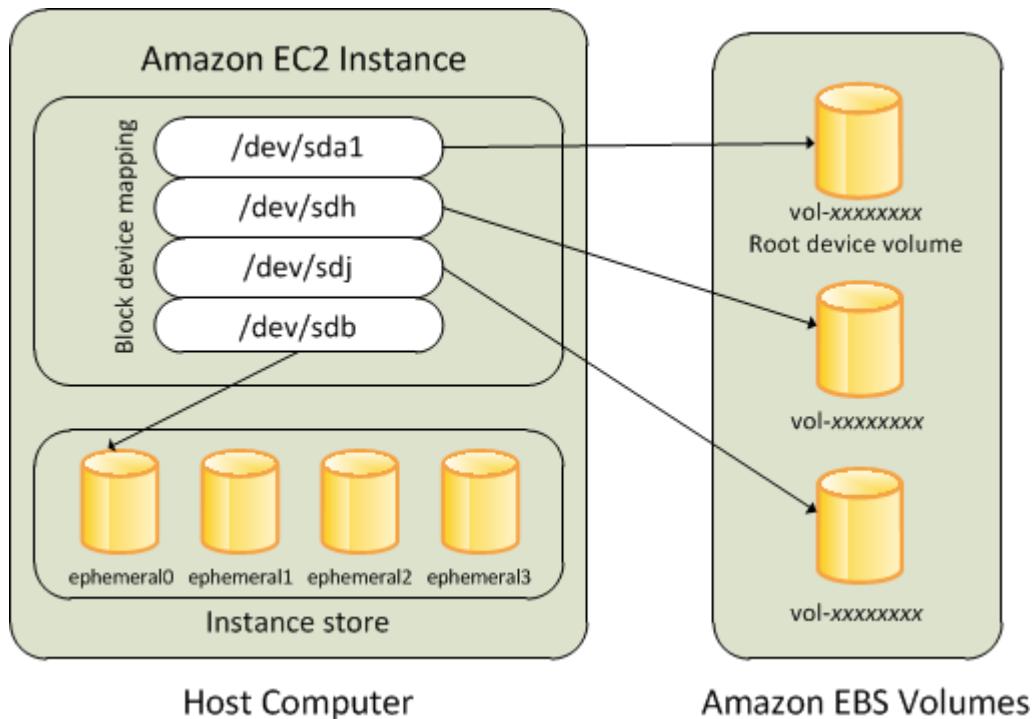
- [Windows] The device name within Amazon EC2, as shown in this table. The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different than the name that Amazon EC2 recommends.

<b>Reserved for the root device</b>	/dev/sda1
<b>Recommended for instance store volumes</b>	xvd[a-e]
<b>Recommended for EBS volumes</b>	xvd[f-p]
<b>Possible for EBS volumes</b>	xvd[a-p] /dev/sda[1-2] /dev/sd[b-e]

- [Instance store volumes only] The virtual device: `ephemeral[0-3]`.
- [EBS volumes only] The ID of the snapshot to use to create the block device (`snap-xxxxxxxx`). This value is optional as long as you specify a volume size.
- [EBS volumes only] The size of the volume, in GiB. The specified size must be greater than or equal to the size of the specified snapshot.
- [EBS volumes only] Whether to delete the volume on instance termination (`true` or `false`). The default value is `true`.
- [EBS volumes only] The volume type (`standard` or `io1`). The default value is `standard`.
- [EBS volumes only] The number of input/output operations per second (IOPS) that the volume supports. (Not used with `standard` volumes.)

## Example Block Device Mapping

This figure shows an example block device mapping for an Amazon EBS-backed instance. It maps `/dev/sdb` to `ephemeral0` and maps two EBS volumes, one to `/dev/sdh` and the other to `/dev/sdj`. It also shows the EBS volume that is the root device volume, `/dev/sda1`.



Note that this example block device mapping is used in the example commands and APIs in this topic. You can find example commands and APIs that create block device mappings here:

- [Specifying a Block Device Mapping for an AMI \(p. 525\)](#)
- [Updating the Block Device Mapping when Launching an Instance \(p. 528\)](#)

## How Devices are Made Available in the Operating System

Device names like `/dev/sdh` and `xvdh` are used by Amazon EC2 to describe block devices. The block device mapping is used by Amazon EC2 to specify the block devices to attach to an Amazon EC2 instance. After a block device is attached to an instance, it must be mounted by the operating system before you can access the storage device. When a block device is detached from an instance, it is unmounted by the operating system and you can no longer access the storage device.

With a Linux instance, the device names specified in the block device mapping are mapped to their corresponding block devices when the instance first boots. The instance type determines which instance store volumes are formatted and mounted by default. You can mount additional instance store volumes at launch, as long as you don't exceed the number of instance store volumes available for your instance type. For more information, see [Amazon EC2 Instance Store \(p. 513\)](#). The block device driver for the instance determines which devices are used when the volumes are formatted and mounted. For more information, see [Attaching a Volume to an Instance \(p. 469\)](#).

With a Windows instance, the device names specified in the block device mapping are mapped to their corresponding block devices when the instance first boots, and then the Ec2Config service initializes and mounts the drives. The root device volume is mounted as `C:\`. The instance store volumes are mounted as `D:\`, `E:\`, and so on. When an EBS volume is mounted, it can be mounted using any available drive letter. However, you can configure how the Ec2Config Service assigns drive letters to EBS volumes; for more information, see [Using EC2Config](#).

## Viewing Block Device Mappings

You can view information about each block device in a block device mapping. For details, see:

- Viewing the EBS Volumes in an AMI Block Device Mapping (p. 527)
- Viewing the EBS Volumes in an Instance Block Device Mapping (p. 530)
- Viewing the Instance Block Device Mapping for Instance Store Volumes (p. 532)

## AMI Block Device Mapping

Each AMI has a block device mapping that specifies the block devices to attach to an instance when it is launched from the AMI. An AMI that Amazon provides includes a root device only. To add additional block devices to an AMI, you must create your own AMI.

## Specifying a Block Device Mapping for an AMI

There are two ways to specify volumes in addition to the root volume when you create an AMI. If you've already attached volumes to a running instance before you create an AMI from the instance, the block device mapping for the AMI includes those same volumes. For EBS volumes, the existing data is saved to a new snapshot and it's this new snapshot that's specified in the block device mapping. For instance store volumes, the data is not preserved.

For an EBS-backed AMI, you can add EBS volumes and instance store volumes using a block device mapping. For an instance store-backed AMI, you can add only instance store volumes using a block device mapping.

## AWS Management Console

### To add volumes to an AMI

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Instances**.
3. Select an instance and select **Create Image (EBS AMI)** from the **Actions** list.
4. In the **Create Image** dialog box, click **Instance Store Volumes**.
5. [Optional] For each instance store volume, select a volume from the **Instance Store** list and a device name from **Device**, and then click **Add**.

Create Image Cancel

Instance ID:	i-9e3815fd BDCTest														
Image Name*:	<input type="text"/>														
Image Description:	<input type="text"/>														
<input type="checkbox"/> No Reboot: <input checked="" type="checkbox"/>															
<input checked="" type="radio"/> Root Volume <input type="radio"/> EBS Volumes <input checked="" type="radio"/> Instance Store Volumes															
Map an instance store volume to the specified device.															
Instance Store: 0 <input type="button" value="▼"/>															
Device:/dev/ <input style="border: none; border-bottom: 1px solid #ccc; padding: 0 2px;" type="button" value="sdb"/>															
<input style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px;" type="button" value="Add"/>															
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Type</th> <th>Device</th> <th>Snapshot ID</th> <th>Size</th> <th>Volume Type</th> <th>IOPS</th> <th>Delete on Termination</th> </tr> </thead> <tbody> <tr> <td>Root</td> <td>/dev/sda1</td> <td></td> <td>8</td> <td>standard</td> <td></td> <td>true</td> </tr> </tbody> </table>		Type	Device	Snapshot ID	Size	Volume Type	IOPS	Delete on Termination	Root	/dev/sda1		8	standard		true
Type	Device	Snapshot ID	Size	Volume Type	IOPS	Delete on Termination									
Root	/dev/sda1		8	standard		true									
<b>0 EBS Volumes 0 Ephemerals</b> Total size of EBS volumes: 8 GB. When you create an EBS image an EBS snapshot will also be created for each of the above volumes.															
<input style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px;" type="button" value="Cancel"/> <input style="border: 1px solid #0070C0; border-radius: 5px; background-color: #0070C0; color: white; padding: 2px 10px;" type="button" value="Yes, Create"/>															

6. Click **Yes, Create**.

## Command Line Interface

Use **ec2-register -b "devicename=blockdevice"** to create an AMI with a block device mapping.

**devicename**

The device name within Amazon EC2

**blockdevice**

To omit a mapping for the device from the AMI, specify `none`.

To add an instance store device, specify `ephemeral[0-3]`.

[EBS-backed instance only] To add an EBS volume, specify `snapshot-id:size:[true|false]`. To add an empty EBS volume, omit the snapshot ID. To indicate whether the EBS volume should be deleted on termination, specify `true` or `false`; the default value is `true`.

For example, run this command at a command prompt to register an Amazon EBS-backed Linux AMI with an EBS root volume, an instance store volume, an EBS volume based on a snapshot, and an empty 100 GiB EBS volume. (Be sure to set the `EC2_PRIVATE_KEY` and `EC2_CERT` environment variables first.)

```
ec2-register -n ImageName --root-device-name /dev/sda1 -s snap-e1eb279f
-b "/dev/sdb=ephemeral0" -b "/dev/sdh=snap-d5eb27ab" -b "/dev/sdj=:100"
```

This command maps `/dev/sda` to an EBS root volume based on a snapshot, `/dev/sdb` to `ephemeral0`, `/dev/sdh` to an EBS volume based on a snapshot, and `/dev/sdj` to an empty EBS volume that is 100 GiB in size. The output is the ID for your new AMI.

```
IMAGE ami-72aa081b
```

For more information, see [ec2-register](#).

To verify that the AMI was created successfully, look for it in the Amazon EC2 Console (click **AMIs** in the navigation pane, then select **Owned By Me** from the **Filter** drop-down list) or the output of the following command:

```
ec2-describe-images -o self
```

Alternatively, if you've already attached volumes to an instance, you can use one of these methods as appropriate.

- Amazon EBS-backed AMI
  - [ec2-create-image](#)
- Instance store-backed AMI
  - [Creating an Instance Store-Backed AMI From an Existing AMI \(p. 36\)](#)
  - [Creating an Instance Store-Backed Windows AMI](#)

## Viewing the EBS Volumes in an AMI Block Device Mapping

You can easily enumerate the EBS volumes in the block device mapping for an AMI.

### AWS Management Console

Use the AWS Management Console as follows to enumerate the EBS volumes in the block device mapping for an AMI:

#### To view the EBS volumes for an AMI

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>
2. In the navigation pane, click **AMIs**.
3. Select **EBS Images** from the **Filter** drop-down list to get a list of EBS-backed AMIs.
4. Locate and the desired AMI and click the **Go to Details Page** button (the magnifying glass). At a minimum, the following information is available for the root device:
  - **Root Device Type** (ebs)
  - **Root Device Name** (for example, /dev/sda1)
  - **Block Devices** (for example, /dev/sda1=snap-e1eb279f:8:true)

If the AMI was created with additional EBS volumes using a block device mapping, the **Block Devices** field displays the mapping for those additional EBS volumes as well. (Recall that this screen doesn't display instance store volumes.)

### Command Line Interface

Use **ec2-describe-images ami\_id** to enumerate the EBS volumes in the block device mapping for an AMI.

For example, run this command at a command prompt to get information about an AMI, including its block device mapping. (Be sure to set the EC2\_PRIVATE\_KEY and EC2\_CERT environment variables first.)

```
ec2-describe-images ami-72aa081b
```

The output includes the block device mapping for your AMI. The following information is available for EBS volumes:

- The device name within Amazon EC2
- The ID of the snapshot used when creating the block device (optional)
- The size of the volume, in GiB
- The volume type.

Here's example output for a Linux AMI.

BLOCKDEVICEMAPPING	/dev/sda1	snap-e1eb279f	8	standard
BLOCKDEVICEMAPPING	/dev/sdh	snap-d5eb27ab	200	standard
BLOCKDEVICEMAPPING	/dev/sdj		100	standard

For more information, see [ec2-describe-images](#).

## Instance Block Device Mapping

By default, an instance that you launch includes any storage devices specified in the block device mapping of the AMI from which you launched the instance. You can specify changes to the block device mapping for an instance when you launch it, and these updates overwrite or merge with the block device mapping of the AMI. However, you can't modify the block device mapping entry for the root device volume.

## Updating the Block Device Mapping when Launching an Instance

You can add EBS volumes and instance store volumes to an instance when you launch it. Note that updating the block device mapping for an instance doesn't make a permanent change to the block device mapping of the AMI from which it was launched.

### AWS Management Console

#### To add volumes to an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the Amazon EC2 console dashboard, click **Launch Instance**.
3. On the **Create a New Instance** page, click **Classic Wizard**.
4. On the **CHOOSE AN AMI** page, choose the AMI to use and click **Select**.
5. Follow the wizard to complete the first and second **INSTANCE DETAIL** pages.
6. On the third **INSTANCE DETAIL** page, click **Edit** under **Storage Device Configuration**. You can modify the root volume, EBS volumes, and instance store volumes as follows:
  - To change the size of the root volume, click **Root Volume**, fill in the **Volume Size** field, and click **Save**.
  - To suppress an EBS volume specified by the block device mapping of the AMI used to launch the instance, click **EBS Volumes**, go to the volume, and click **Delete**.
  - To add an EBS volume, click **EBS Volumes**, fill in the fields (**Device**, **Snapshot**, and so on), and click **Add**.
  - To suppress an instance store volume specified by the block device mapping of the AMI used to launch the instance, click **Instance Store Volumes**, go to the volume, and click **Delete**.
  - To add an instance store volume, click **Instance Store Volumes**, select the **Instance Store** and **Device**, and click **Add**.

**Request Instances Wizard**

CHOOSE AN AMI    INSTANCE DETAILS    CREATE KEY PAIR    CONFIGURE FIREWALL    REVIEW    Cancel

**Number of Instances:** 1

**Availability Zone:** No Preference

**Storage Device Configuration**

Your instance will be launched with the following storage device settings. Edit these settings to add EBS volumes, instance store volumes, or edit the settings of the root volume.

Root Volume     EBS Volumes     Instance Store Volumes

Map an instance store volume to the specified device.

Instance Store: 0

Device: /dev/ sdb

Type	Device	Snapshot ID	Size	Volume Type	IOPS	Delete on Termination
Root	/dev/sda1	snap-495fcc0e	8	standard		true

**0 EBS Volumes    0 Ephemerals**

[Back](#)    [Continue](#)

7. Complete the remaining wizard pages, and then click **Launch Instance**.

## Command Line Interface

Use **ec2-run-instances -b "devicename=blockdevice"** to define an entry for the block device mapping for an instance.

### **devicename**

The device name within Amazon EC2

### **blockdevice**

To omit a mapping for the device from the AMI, specify `none`.

To add an instance store device, specify `ephemeral[0-3]`.

[EBS-backed instance only] To add an EBS volume, specify `snapshot-id:size:[true|false]`. To add an empty EBS volume, omit the snapshot ID. To indicate whether the EBS volume should be deleted on termination, specify `true` or `false`; the default value is `true`.

For example, suppose that an EBS-backed Linux AMI specifies the following block device mapping:

- `/dev/sdb=ephemeral0`
- `/dev/sdh=snap-92d333fb`
- `/dev/sdj=:100`

To prevent `/dev/sdj` from attaching to an instance launched from this AMI, use this option.

```
-b "/dev/sdj=none"
```

To increase the size of /dev/sdh to 300 GiB, use this option.

```
-b "/dev/sdh=:300"
```

Notice that we didn't need to specify the snapshot ID for /dev/sdh, because specifying the device name is enough to identify the volume.

To attach an additional instance store volume, /dev/sdc, use this option. If the instance type doesn't support more than one instance store volume, this option has no effect.

```
-b "/dev/sdc=ephemeral1"
```

For more information, see [ec2-run-instances](#).

## Viewing the EBS Volumes in an Instance Block Device Mapping

You can easily enumerate the EBS volumes mapped to an instance.

### Note

For instances launched before the release of the 2009-10-31 API, AWS can't display the block device mapping. You must detach and reattach the volumes so that AWS can display the block device mapping.

### AWS Management Console

Use the AWS Management Console to enumerate the EBS volumes in the block device mapping for an instance.

#### To view the EBS volumes for an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>
2. In the navigation pane, click **Instances**.
3. In the **Viewing** menu, click **EBS Root Device** to filter the list. This displays a list of Amazon EBS-backed instances.
4. Locate and click the desired instance and look at the details displayed in the **EC2 Instances** pane. At a minimum, the following information is available for the root device:
  - **Root Device Type** (ebs)
  - **Root Device** (for example, sda1)
  - **Block Devices** (for example, sda1, sdh, and sdj)

If the instance was launched with additional EBS volumes using a block device mapping, the **Block Devices** box displays those additional EBS volumes as well as the root device. (Recall that this dialog box doesn't display instance store volumes.)

Description	Status Checks	Monitoring	Tags
AMI: amzn-ami-2011.09.1.x86_64-ebs (ami-7341831a)			
Zone: us-east-1c			
Type: m1.small			
Scheduled Events: No scheduled events			
VPC ID: -			
Source/Dest. Check: -			
Placement Group: -			
RAM Disk ID: -			
Key Pair Name: -			
Monitoring: basic			
Elastic IP: -			
<b>Root Device Type:</b> <b>ebs</b>			
IAM Role: -			
Lifecycle: normal			
<b>Block Devices:</b> <b>sda1</b> sdh sdj			

**Network Interfaces:**

- To display additional information about a block device, click its entry under Block Devices. This displays the following information for the block device:
  - EBS ID (vol-xxxxxxx)
  - Root Device Type** (ebs)
  - Attachment time** (yyyy-mmThh:mm:ss.ssTZD)
  - Block device status** (attaching, attached, detaching, detached)
  - Delete on termination** (Yes, No)
  - Snapshot ID** (snap-xxxxxxx)

<b>sda1</b>	
<b>EBS ID:</b>	vol-7171301f
<b>Root device type:</b>	ebs
<b>Attachment time:</b>	2012-06-19T15:24:21.000Z
<b>Block device status:</b>	attached
<b>Delete on termination:</b>	Yes
<b>Snapshot ID:</b>	snap-8d3fd3ee

## Command Line Interface

Use **ec2-describe-instances** *instance\_id* to enumerate the EBS volumes in the block device mapping for an instance.

For example, run this command at a command prompt to get information about an instance, including its block devices. (Be sure to set the EC2\_PRIVATE\_KEY and EC2\_CERT environment variables first.)

```
ec2-describe-instances i-xxxxxx
```

The output includes the block devices for your instance. The following information is available for EBS volumes:

- The device name within Amazon EC2
- The volume ID
- The time when the attachment was initiated
- Whether to delete the volume on instance termination
- The volume type (standard or io1)
- The number of input/output operations per second (IOPS) that the volume supports. (Not used with standard volumes.)

The output for a Linux instance looks something like this.

BLOCKDEVICE	/dev/sda1	vol-xxxxxxxx	YYYY-mmThh:mm:ss.sTZD	true
BLOCKDEVICE	/dev/sdh	vol-xxxxxxxx	YYYY-mmThh:mm:ss.sTZD	true
BLOCKDEVICE	/dev/sdj	vol-xxxxxxxx	YYYY-mmThh:mm:ss.sTZD	true

To view this information plus the status (attaching, attached, detaching, detached), run the **ec2-describe-instances** command with the **-v** option to see the full response.

For more info, see [ec2-describe-instances](#).

## Viewing the Instance Block Device Mapping for Instance Store Volumes

When you view the block device mapping for your instance, you can see only the EBS volumes, not the instance store volumes. You can use instance metadata to query the complete block device mapping. The base URI for all requests for instance metadata is <http://169.254.169.254/latest/>.

First, connect to your running instance. If the instance is running Linux, the GET command is already available. If the instance is running Windows, install wget on the instance, and replace GET with wget in the examples below.

Use this query on a running instance to get its block device mapping.

```
GET http://169.254.169.254/latest/meta-data/block-device-mapping/
```

The response includes the names of the block devices for the instance. For example, the output for an instance store-backed m1.small instance looks like this.

```
ami
ephemeral0
root
swap
```

The ami device is the root device as seen by the instance. The instance store volumes are named ephemeral[0-3]. The swap device is for the page file. If you've also mapped EBS volumes, they appear as ebs1, ebs2, and so on.

To get details about an individual block device in the block device mapping, append its name to the previous query, as shown here.

```
GET http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

For more info, see [Instance Metadata and User Data \(p. 365\)](#).

# Resources and Tags

---

The following topics describe features that enable you to manage Amazon EC2 resources, such as AMIs, instances, Amazon EBS volumes, and snapshots.

## Topics

- [Resource Locations \(p. 534\)](#)
- [Listing and Filtering Your Resources \(p. 535\)](#)
- [Tagging Your Amazon EC2 Resources \(p. 540\)](#)

## Resource Locations

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

Resource	Type	Description
AWS Account	Global	You can use the same AWS account in all regions.
DevPay Product Codes	Global	You can use the same DevPay product codes in all regions.
SSH Key Pairs	Global or Regional	<p>You can use the SSH key pairs that you create using Amazon EC2 only in the region where you created them. For more information, see <a href="#">How to Have AWS Create the Key Pair for You (p. 275)</a>.</p> <p>You can create and upload an RSA key pair that you can use in all regions. For more information, see <a href="#">How to Generate Your Own Key and Import It to Amazon EC2 (p. 279)</a>.</p>
Amazon EC2 Resource Identifiers	Regional	Each resource identifier, such as an AMI ID, instance ID, EBS volume ID, or EBS snapshot ID, is tied to its region and can be used only in the region where you created the resource.

Resource	Type	Description
User-Supplied Resource Names	Regional	Each resource name, such as a security group name or an SSH key pair name, is tied to its region and can be used only in the region where you created the resource. Although you can create resources with the same name in multiple regions, they aren't related to each other.
AMIs	Regional	An AMI is tied to the region where its files are located within Amazon S3. You can copy an AMI from one region to another. For more information, see <a href="#">Copying AMIs (p. 67)</a> .
Elastic IP Addresses	Regional	An Elastic IP address is tied to a region and can be associated only with an instance in the same region.
Security Groups	Regional	A security group is tied to a region and can be assigned only to instances in the same region. You can't enable an instance to communicate with an instance outside its region using security group rules. Traffic from an instance in another region is seen as WAN bandwidth.
EBS Snapshots	Regional	An EBS snapshot is tied to its region and can only be used to create volumes in the same region. You can copy a snapshot from one region to another. For more information, see <a href="#">Copying an Amazon EBS Snapshot (p. 494)</a> .
EBS Volumes	Availability Zone	An Amazon EBS volume is tied to its Availability Zone and can be attached only to instances in the same Availability Zone.
Instances	Availability Zone	An instance is tied to the Availability Zones in which you launched it. However, note that its instance ID is tied to the region.

#### Related Topic

- [Regions and Availability Zones \(p. 116\)](#)

## Listing and Filtering Your Resources

As part of using Amazon EC2, you might use different *resources*. These include images, instances, Amazon EBS volumes, and snapshots. You can get a list of some types of resource using the Amazon EC2 console. You can get a list of each type of resource using its corresponding command or API action. For example, you can list Amazon Machine Images (AMI) using `ec2-describe-images` or `DescribeImages`, and you can list instances using `ec2-describe-instances` or `DescribeInstances`.

#### Topics

- [Filters \(p. 536\)](#)
- [Listing Resources \(p. 536\)](#)
- [Filtering Resources \(p. 537\)](#)

## Filters

The resulting lists of resources can be long, so you might want to filter the results to include only the resources that match certain criteria. For example, you can list only the public 64-bit Windows AMIs that use an Amazon EBS volume as the root device volumes. As another example, you can list only the Amazon EBS snapshots that have been tagged with a certain value.

You can specify multiple filter values. For example, you can list all the instances whose type is either `m1.small` or `m1.large`. The resource must match at least one of the values to be included in the resulting resource list. You can also specify multiple filters. For example, you can list all the instances whose type is either `m1.small` or `m1.large`, and that have an attached EBS volume that is set to delete when the instance terminates. The instance must match all your filters to be included in the results.

You can also use wildcards with the filter values. An asterisk (\*) matches zero or more characters, and a question mark (?) matches exactly one character. For example, you can use `*database*` as a filter value to get all EBS snapshots that include `database` in the description. If you were to specify `database` as the filter value, then only snapshots whose description equals `database` would be returned. Filter values are case sensitive. We support only exact string matching, or substring matching (with wildcards).

**Tip**

Your search can include the literal values of the wildcard characters; you just need to escape them with a backslash before the character. For example, a value of `\*amazon\?\\\` searches for the literal string `*amazon?\\`.

## Listing Resources

You can list your resources using the console, command-line interface, or API actions.

### AWS Management Console

You can view the most common EC2 resource types using the AWS Management Console. To view additional resources, use the command line interface or the API actions.

#### To list EC2 resources using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click the option that corresponds to the resource, such as **AMIs** or **Instances**.

**EC2 Dashboard**

Events  
Tags

**INSTANCES**

Instances  
Spot Requests  
Reserved Instances

**IMAGES**

AMIs  
Bundle Tasks

**ELASTIC BLOCK STORE**

Volumes  
Snapshots

**NETWORK & SECURITY**

Security Groups  
Elastic IPs  
Placement Groups  
Load Balancers  
Key Pairs  
Network Interfaces

## Command Line Interface

Each resource type has a corresponding command that you use to list resources of that type. For example, you can use the `ec2-describe-images` command to list your images, and the `ec2-describe-instances` command to list your instances.

The following example uses `ec2-describe-instances` to list your instances.

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

The response contains information for all your instances.

## API

Each resource type has a corresponding API action that you use to list resources of that type. For example, you can use the `DescribeImages` action to list your images, and the `DescribeInstances` action to list your instances.

The following example uses `DescribeInstances` to list your instances.

```
https://ec2.amazonaws.com/?Action=DescribeInstances
&AuthParams
```

The response contains information for all your instances.

## Filtering Resources

You can filter a list of your resources using the console, command-line interface, or API actions.

## AWS Management Console

You can perform some basic filtering and sorting of the most common resource types using the AWS Management Console. For additional filtering capabilities, use the command line interface or the API actions.

### To list volumes in the us-east-1b Availability Zone with a status of available

1. In the navigation pane, click **Volumes**.
2. In the **Viewing** pane, select **Detached Volumes** from the drop-down list. (A detached volume is available to be attached to an instance in the same Availability Zone.)
3. In the **Viewing** pane, enter `us-east-1b` in the search field.
4. Any volumes that meet this criteria are displayed.

The screenshot shows a table with the following columns: Name, Volume ID, Capacity, Volume Type, Snapshot, Created, and Zone. There is one row displayed, representing a volume named 'empty' with Volume ID 'vol-56d9b325', capacity '100 GiB', type 'standard', snapshot '--', created date '2013-04-02T1', and located in 'us-east-1b'.

Name	Volume ID	Capacity	Volume Type	Snapshot	Created	Zone
empty	vol-56d9b325	100 GiB	standard	--	2013-04-02T1	us-east-1b

### To list public 64-bit Windows AMIs backed by Amazon EBS

1. In the navigation pane, click **AMIs**.
2. In the **Filter** pane, select **Public Images**, **EBS Images** and **Windows** from the drop-down lists.
3. Enter `x86_64` in the search field.
4. Any AMIs that meet this criteria are displayed.

The screenshot shows a table with the following columns: Launch, Actions, Filter: Public Images, EBS Images, Windows, and a search bar containing 'x86\_64'. The table lists six AMIs, all of which are .NET Beanstalk AMIs with various IDs and creation dates, all marked as Public and available, and running on Windows.

Launch	Actions	Filter: Public Images	EBS Images	Windows	x86_64	X

## Command Line Interface

You can add one or more filters to an `ec2-describe-XXX` command to scope the results to resources that match certain criteria. For a list of the available filters for a given EC2 resource, go to the `ec2-describe-XXX` topic for that resource in the [Amazon Elastic Compute Cloud Command Line Reference](#).

### Tip

If you're using the command line tools on Windows, you might need to use quotation marks as shown here.

The following example uses filters so that [ec2-describe-volumes](#) lists only volumes that are in the us-east-1b Availability Zone and have a status of available.

```
PROMPT> ec2-describe-volumes --filter "availability-zone=us-east-1b" --filter "status=available"
```

The following example uses filters so that [ec2-describe-images](#) lists only public 64-bit Windows AMIs that use an Amazon EBS volume as the root device.

```
PROMPT> ec2-describe-images --filter "is-public=true" --filter "architecture=x86_64" --filter "platform=windows"
```

The following example uses filters so that [ec2-describe-instances](#) lists only your m1.small or m1.large instances that have an attached Amazon EBS volume that is set to delete on instance termination.

```
PROMPT> ec2-describe-instances --filter "instance-type=m1.small" --filter "instance-type=m1.large" --filter "block-device-mapping.status=attached" --filter "block-device-mapping.delete-on-termination=true"
```

For an example of filtering resources by tags, see [Tagging Your Amazon EC2 Resources \(p. 540\)](#).

## API

You can add one or more filters to a `DescribeXXX` action to scope the results to resources that match certain criteria. For a list of the available filters for a given EC2 resource, go to the `DescribeXXX` topic for that resource in the [Amazon Elastic Compute Cloud API Reference](#).

The following example request uses filters with [DescribeVolumes](#) to list only your volumes that are in the us-east-1b Availability Zone and have a status of available.

```
https://ec2.amazonaws.com/?Action=DescribeVolumes
&Filter.1.Name=availability-zone
&Filter.1.Value.1=us-east-1b
&Filter.2.Name=status
&Filter.2.Value.1=available
&AUTHPARAMS
```

The following example request uses filters with [DescribeImages](#) to list only public 64-bit Windows AMIs that use an Amazon EBS volume as the root device.

```
https://ec2.amazonaws.com/?Action=DescribeImages
&Filter.1.Name=is-public
&Filter.1.Value.1=true
&Filter.2.Name=architecture
&Filter.2.Value.1=x86_64
&Filter.3.Name=platform
&Filter.3.Value.1=windows
&AUTHPARAMS
```

The following example request uses filters with [DescribeInstances](#) to list only the m1.small or m1.large instances that have an attached Amazon EBS volume that is set to delete on instance termination.

```
https://ec2.amazonaws.com/?Action=DescribeInstances
&Filter.1.Name=instance-type
&Filter.1.Value.1=m1.small
&Filter.1.Value.2=m1.large
&Filter.2.Name=block-device-mapping.status
&Filter.2.Value.1=attached
&Filter.3.Name=block-device-mapping.delete-on-termination
&Filter.3.Value.1=true
&AUTHPARAMS
```

For an example of filtering resources by tags, see [Tagging Your Amazon EC2 Resources \(p. 540\)](#).

## Tagging Your Amazon EC2 Resources

To help you manage your instances, images, and other Amazon Elastic Compute Cloud (Amazon EC2) resources, you can assign your own metadata to each resource in the form of *tags*. This topic describes tags and shows you how to create them.

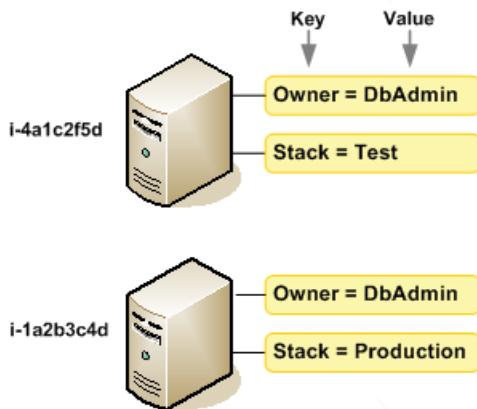
### Topics

- [Tag Basics \(p. 540\)](#)
- [Tag Restrictions \(p. 541\)](#)
- [Tagging Your Resources for Billing \(p. 542\)](#)
- [Using the AWS Management Console \(p. 543\)](#)
- [Using the Command Line Interface \(p. 552\)](#)
- [Using the API \(p. 555\)](#)

## Tag Basics

Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. Each tag consists of a key and a value, both of which you define. For example, you could define a set of tags for your account's Amazon EC2 instances that helps you track each instance's owner and stack level. We recommend that you devise a set of tag keys that meets your needs for each resource type. Using a consistent set of tag keys makes it easier for you to manage your resources. You can search and filter the resources based on the tags you add.

The following diagram illustrates how tagging works. In this example, you've assigned two tags to each of your instances, one called `Owner` and another called `Stack`. Each of the tags also has an associated value.



Tags don't have any semantic meaning to Amazon EC2 and are interpreted strictly as a string of characters. Also, tags are not automatically assigned to your resources.

You can work with tags using the AWS Management Console, the Amazon EC2 command line interface (CLI), and the Amazon EC2 API.

You can assign tags only to resources that already exist. When you use the Amazon EC2 console, you can access a list of tags to add to an instance, which will be applied immediately after the instance is created. If you add a tag that has the same key as an existing tag on that resource, the new value overwrites the old value. You can edit tag keys and values, and you can remove tags from a resource at any time. You can set a tag's value to the empty string, but you can't set a tag's value to null.

If you're using AWS Identity and Access Management (IAM), you can control which users in your AWS account have permission to create, edit, or delete tags. For more information about IAM, see [Controlling Access to Amazon EC2 Resources \(p. 421\)](#).

## Tag Restrictions

The following basic restrictions apply to tags:

- Maximum number of tags per resource—10
- Maximum key length—128 Unicode characters
- Maximum value length—256 Unicode characters
- Tag keys and values are case sensitive.
- Do not use the `aws:` prefix in your tag names or values because it is reserved for AWS use. You can't edit or delete tag names or values with this prefix.

You can't terminate, stop, or delete a resource based solely on its tags; you must specify the resource identifier. For example, to delete snapshots that you tagged with a tag key called `DeleteMe`, you must first get a list of those snapshots using `DescribeSnapshots` with a filter that specifies the tag. Then you use `DeleteSnapshots` with the resource identifiers of the snapshots (for example, `snap-1a2b3c4d`). You can't call `DeleteSnapshots` with a filter that specified the tag. For more information about using filters when listing your resources, see [Listing and Filtering Your Resources \(p. 535\)](#).

You can tag public or shared resources, but the tags you assign are available only to your AWS account and not to the other accounts sharing the resource.

You can't tag all resources, and some you can only tag using API actions or the command line. The following table lists all Amazon EC2 resources and the tagging restrictions that apply to them, if any. Resources with tagging restrictions of None can be tagged with API actions, the CLI, and the console.

Resource	Tagging support	Tagging restrictions
AMI	Yes	None
Bundle Task	No	
Customer Gateway	Yes	None
DHCP Option	Yes	None
EBS Volume	Yes	None
Elastic IP	No	
Instance	Yes	None
Internet Gateway	Yes	None
Key Pair	No	
Load Balancer	No	
Network ACL	Yes	None
Network Interface	Yes	None
Placement Group	No	
Reserved Instance	Yes	None
Reserved Instance Listing	Yes	None
Route Table	Yes	None
Spot Instance Request	Yes	Tag with CLI and API only
Security Group - EC2 Classic	Yes	None
Security Group - VPC	Yes	None
Snapshot	Yes	None
Subnet	Yes	None
Virtual Private Gateway	Yes	None
VPC	Yes	None
VPN Connection	Yes	None

For more information about tagging using the AWS console, see [Using the AWS Management Console \(p. 543\)](#). For more information about tagging using API actions, see [Using the API \(p. 555\)](#). For more information about tagging using the command line, see [Using the Command Line Interface \(p. 552\)](#).

## Tagging Your Resources for Billing

You can use tags to organize your AWS bill to reflect your own cost structure. To do this, sign up to get your AWS account bill with tag key values included. Then, to see the cost of your combined resources,

you can organize your billing information based on resources that have the same tag key values. For example, you can tag several resources with a specific application name, and then organize your billing information to see the total cost of that application across several services. For more information, see [Cost Allocation and Tagging](#) in *About AWS Account Billing*.

## Using the AWS Management Console

Using the Amazon EC2 console, you can see which tags are in use across all of your Amazon EC2 resources in the same region. You can view tags by resource and by resource type, and you can also view how many items of each resource type are associated with a specified tag. You can also use the Amazon EC2 console to apply or remove tags from one or more resources at a time.

### Topics

- [Displaying Tags \(p. 543\)](#)
- [Adding and Deleting Tags on an Individual Resource \(p. 544\)](#)
- [Adding and Deleting Tags to a Group of Resources \(p. 546\)](#)
- [Adding a Tag When You Launch an Instance \(p. 548\)](#)
- [Filtering a List of Resources by Tag \(p. 551\)](#)

## Displaying Tags

You can display tags in two different ways in the Amazon EC2 console. You can display the tags for an individual resource or for all resources.

### To display tags for individual resources

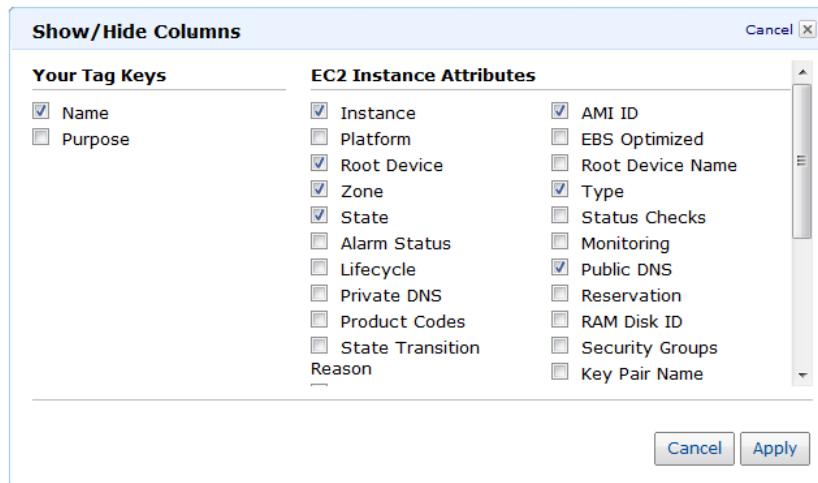
When you select a resource-specific page in the Amazon EC2 console, it displays a list of those resources. For example, if you select **Instances** from the navigation pane, the console displays a list of Amazon EC2 instances. When you select a resource from one of these lists (e.g., an instance), if the resource supports tags, you can view and manage its tags. Depending on which resource page you're on, you can view the tags either in the **Tags** tab on the details pane, or by clicking **Go to Details Page** (the magnifying glass). The following image shows the **Tags** tab for an instance with two tags: Name = DNS Server and Purpose = Network Management.

Add tags to your instance to simplify the administration of your EC2 infrastructure. A form of metadata, tags consist of a case-sensitive key/value pair, are stored in the cloud and are private to your account. You can create user-friendly names that help you organize, search, and browse your resources. For example, you could define a tag with key = Name and value = Webserver. You can add up to 10 unique keys to each instance along with an optional value for each key. For more information, go to [Tagging Your Amazon EC2 Resources](#) in the *EC2 User Guide*.

Key	Value	Actions
Name	DNS Server	Remove Tag    Show Column
Purpose	Network Management	Remove Tag    Hide Column

You can add a column to the resource list that displays all values for tags with the same key. This column enables you to sort and filter the resource list by the tag. There are two ways to add a new column to the resource list to display your tags.

- On the **Tags** tab, click **Show Column** for the tag.
- Click the **Show/Hide Columns** gear-shaped icon, and in the **Show/Hide Columns** dialog box, select the tag key under **Your Tag Keys**, and then click **Apply**.



### To display tags for all resources

You can display tags across all resources by selecting **Tags** from the navigation pane in the Amazon EC2 console. The following image shows the **Tags** pane, which lists all tags in use by resource type.

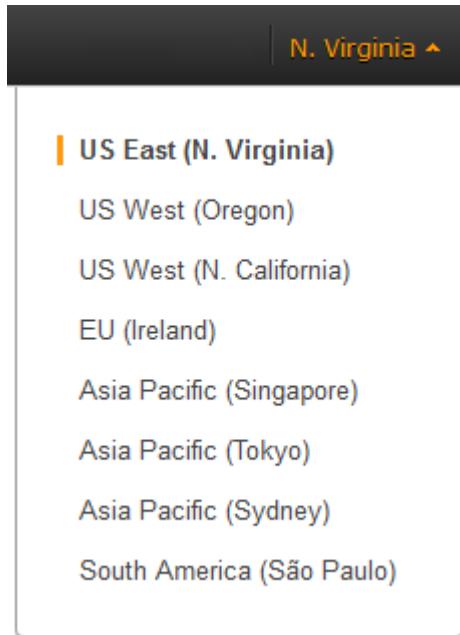
Manage Tags						
Filter: <input type="text"/> Search Keys <input type="button" value="X"/>		Search Values <input type="button" value="X"/>		1 to 7 of 7 Tags		
	Tag Key	Tag Value	Total	Instances	AMIs	Volumes
Manage Tag	Name	DNS Server	1	1	0	0
Manage Tag	Owner	TeamB	2	0	0	2
Manage Tag	Owner	TeamA	2	0	0	2
Manage Tag	Purpose	Project2	1	0	0	1
Manage Tag	Purpose	Logs	1	0	0	1
Manage Tag	Purpose	Network Management	1	1	0	0
Manage Tag	Purpose	Project1	2	0	0	2

## Adding and Deleting Tags on an Individual Resource

You can manage tags for an individual resource directly from the resource's page. If you are managing an AMI's tags, the procedures are different from that of other resources. All procedures are explained below.

### To add a tag to an individual resource

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations \(p. 534\)](#).



3. In the navigation pane, click a resource type (for example, **Instances**).
4. Select the resource from the resource list.
5. Select the **Tags** tab in the details pane.
6. Click the **Add/Edit Tags** button.
7. In the **Tag EC2 Instance** dialog box, specify the key and value for each tag, and then click **Save Tags**. Notice that the dialog box automatically suggests **Name** as the key for the first tag; however, you can enter a different key.

The dialog box has a title bar 'Tag EC2 Instance' with a 'Cancel' button. The main area contains a descriptive text about tags and their purpose. Below the text is a table for adding tags. The table has three columns: 'Key' (127 characters maximum), 'Value' (255 characters maximum), and 'Remove'. There are two rows in the table, both labeled 'Name' in the 'Key' column and have red 'X' icons in the 'Remove' column. At the bottom of the table is a link 'Add another Tag. (Maximum of 10)'. At the very bottom are 'Cancel' and 'Save Tags' buttons.

Key (127 characters maximum)	Value (255 characters maximum)	Remove
Name		X
		X

### To add a tag to an AMI

1. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations \(p. 534\)](#).
2. In the navigation pane, click **AMIs**.
3. In the row of the AMI you want to add the tag to, click the **Go to Details Page** button (the magnifying glass).

4. Expand the **Tags** area, and click **Create Tag**.
5. Specify the key and value for the tag, and then click the **Save** checkmark icon. To add more tags, click **Create Tag** and repeat this step.

#### To delete a tag from an individual resource

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations \(p. 534\)](#).



3. In the navigation pane, click a resource type (for example, **Instances**).
4. Select the resource from the resource list.
5. Select the **Tags** tab in the details pane.
6. Click **Remove Tag** in the **Actions** column for the tag.

#### To delete a tag from an AMI

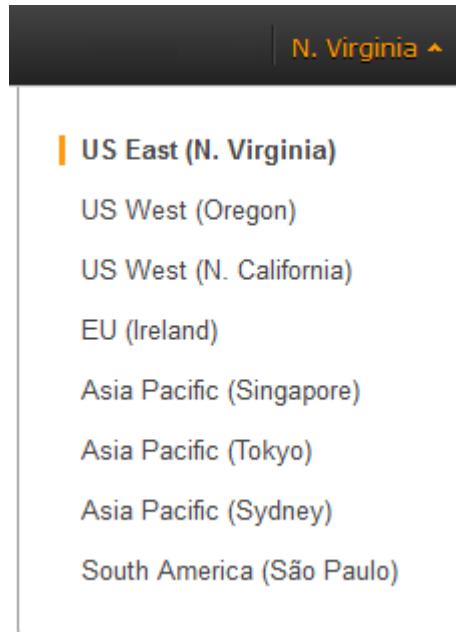
1. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations \(p. 534\)](#).
2. In the navigation pane, click **AMIs**.
3. In the row of the AMI you want remove the tag from, click the **Go to Details Page** button (the magnifying glass).
4. Expand the **Tags** area, and click the **Delete** button for the tag.

## Adding and Deleting Tags to a Group of Resources

#### To add a tag to a group of resources

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

- From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations \(p. 534\)](#).



- In the navigation pane, click **Tags**.
- At the top of the content pane, click **Manage Tags**.
- From the **Filter** drop-down list, select the type of resource (for example, instances) that you want to add tags to.

Add tags to your resources to simplify the administration of your EC2 infrastructure. Select resources from the grid below and use the controls below the grid to apply a new tag or remove an existing tag. You can add up to 10 unique keys to each resource with an optional value for each key. Tag keys and values are case-sensitive.

Filter: Instances  1 to 2 of 2 Resources

Resource ID	Owner	Name
i-04202377		
i-6f8fb013		DNS Server

Add Tag

Key  Value  Add Tag

OR

Remove Tag

Key  Remove Tag

6. In the resources list, select the check box next to each resource that you want to add tags to.
7. In the **Key** and **Value** boxes under **Add Tag**, type the tag key and values you want, and then click **Add Tag**.

**Note**

If you add a new tag with the same tag key as an existing tag, the new tag overwrites the existing tag.

### To remove a tag from a group of resources

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations \(p. 534\)](#).

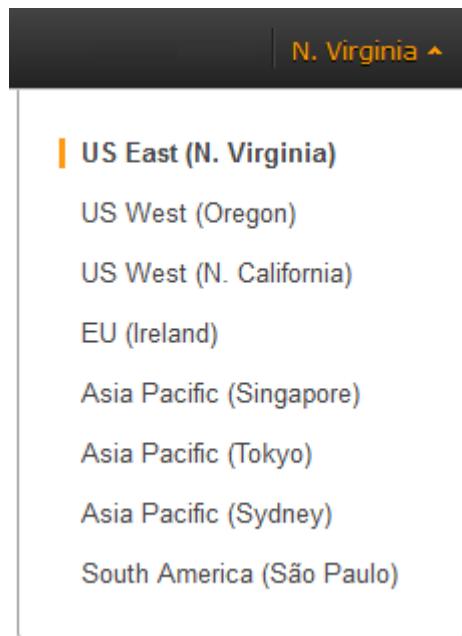


3. In the navigation pane, click **Tags**.
4. At the top of the content pane, click **Manage Tags**.
5. To view the tags in use, click the **Show/Hide Columns** gear-shaped icon, and in the **Show/Hide Columns** dialog box, select the tag keys you want to view, and then click **Close**.
6. From the **Filter** drop-down list, select the type of resource (for example, instances) that you want to remove tags from.
7. In the resource list, select the check box next to each resource that you want to remove tags from.
8. Under **Remove Tag**, click in the **Key** box to select a key, or type its name, and then click **Remove Tag**.

## Adding a Tag When You Launch an Instance

### To add a tag using the Classic Wizard

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region for the instance. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. Select the region that meets your needs. For more information, see [Resource Locations \(p. 534\)](#).



3. Click the **Launch Instance** button on the EC2 dashboard. On the **Create a New Instance** page, select **Classic Wizard**, and then click **Continue**.
4. On the **CHOOSE AN AMI** page, the **Quick Start** tab displays a list of basic configurations called Amazon Machine Images (AMI). Choose the AMI that you want to use and click its **Select** button. For more information about selecting an AMI, see [Finding a Suitable AMI \(p. 21\)](#).
5. Follow the directions in the wizard for the initial **INSTANCE DETAILS** pages.
6. When prompted, specify the key and value for each tag, and then click **Continue**. Notice that the wizard automatically suggests `Name` as the key for the first tag; however, you can enter a different key.

Request Instances Wizard

Cancel 

CHOOSE AN AMI    INSTANCE DETAILS    CREATE KEY PAIR    CONFIGURE FIREWALL    REVIEW

Add tags to your instance to simplify the administration of your EC2 infrastructure. A form of metadata, tags consist of a case-sensitive key/value pair, are stored in the cloud and are private to your account. You can create user-friendly names that help you organize, search, and browse your resources. For example, you could define a tag with key = Name and value = Webserver. You can add up to 10 unique keys to each instance along with an optional value for each key. For more information, go to [Tagging Your Amazon EC2 Resources](#) in the *EC2 User Guide*.

Key (127 characters maximum)	Value (255 characters maximum)	Remove
Name		
		

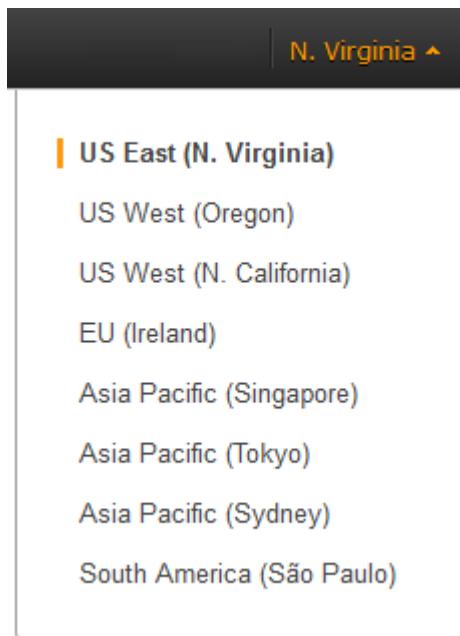
Add another Tag. (Maximum of 10)

< Back Continue 

7. On the **CREATE A KEY PAIR** page, you can choose from any existing key pairs that you've created, or follow the wizard directions to create a new key pair, and then click **Continue**.
8. On the **Configure Firewall** page, you can choose from any existing security group that you own, or follow the wizard directions to create a new security group, and then click **Continue**.
9. Review your settings. When you're satisfied with your selections, click **Launch**.

#### To add a tag using the Quick Launch Wizard

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations \(p. 534\)](#).



3. Click the **Launch Instance** button from the dashboard. On the **Create a New Instance** page, select **Quick Launch Wizard**, and then click **Continue**.
4. Specify a name for your instance, choose a key pair, select an AMI, and then click **Continue**.
5. Click **Edit details** and select **Modify Tags**.
6. Specify the key and value for each tag, and then click **Save details**.

Modify Tags		
Use tags to add information about your instance. For more information, see <a href="#">Using Tags</a> in the EC2 User Guide.		
Key (127 characters maximum)	Value (255 characters maximum)	Remove
<input type="text"/>	<input type="text"/>	
<a href="#">Add another Tag. (Maximum of 10)</a>		

7. Review your settings. When you're satisfied with your selections, click **Launch**.

## Filtering a List of Resources by Tag

You can filter your list of resources based on one or more tag keys and tag values. Note this is currently not possible on the AMIs resource page.

### To filter a list of resources by tag

1. Display a column for the tag as follows:
  - a. Select one of the resources.
  - b. Select the **Tags** tab in the details pane.,
  - c. Locate the tag in the list and click **Show Column**.
2. Click the filter icon in the top right corner of the column for the tag to display the filter list.

	Owner	Purpose	Zone	State
<input type="checkbox"/>	TeamA	Project1		
<input type="checkbox"/>	TeamA	Project2		
<input type="checkbox"/>	TeamB	Logs		
<input type="checkbox"/>	TeamB	Project1		

**0 Volumes selected**  
 Select a volume above

3. Select the tag values, and then click **Apply Filter** to filter the results list.

	Owner	Purpose	Zone	State
<input type="checkbox"/>	TeamA	Project1	us-east-1b	<span style="color: blue;">available</span>
<input type="checkbox"/>	TeamB	Project1	us-east-1c	<span style="color: green;">in-use</span>

## Using the Command Line Interface

Use the following CLI commands to add, update, list, and delete the tags for your resources. The documentation for each command provides examples.

Description	Command
Adds or overwrites one or more tags for the specified resource or resources.	<a href="#">ec2-create-tags</a>
Deletes the specified tags from the specified resource or resources.	<a href="#">ec2-delete-tags</a>
Describes one or more tags for your resources.	<a href="#">ec2-describe-tags</a>

In the following sections, you can learn how to use the CLI to list tags, or list resources and filter the results based on tags.

### Topics

- [Adding Tags \(p. 553\)](#)
- [Listing Tags \(p. 553\)](#)
- [Changing Tag Values \(p. 554\)](#)
- [Deleting Tags \(p. 554\)](#)
- [Filtering the List of Resources by Tags \(p. 554\)](#)

## Adding Tags

The following example adds the same two tags to an AMI and an instance. For the first tag, the key is `webserver` and there is no value. For the other tag, the key is `stack` and the value is `Production`.

```
PROMPT> ec2-create-tags ami-1a2b3c4d i-6f5d4e3a --tag webserver --tag stack=Production
TAG ami-1a2b3c4d image webserver
TAG ami-1a2b3c4d image stack Production
TAG i-6f5d4e3a image webserver
TAG i-6f5d4e3a image stack Production
```

## Listing Tags

You can list your tags and filter the results in different ways, as shown in the following examples.

This example describes all your tags.

```
PROMPT> ec2-describe-tags
TAG ami-1a2b3c4d image webserver
TAG ami-1a2b3c4d image stack Production
TAG i-5f4e3d2a instance webserver
TAG i-5f4e3d2a instance stack Production
TAG i-12345678 instance database_server
TAG i-12345678 instance stack Test
```

This example describes the tags for your resource with ID `ami-1a2b3c4d`.

```
PROMPT> ec2-describe-tags --filter "resource-id=ami-1a2b3c4d"
TAG ami-1a2b3c4d image webserver
TAG ami-1a2b3c4d image stack Production
```

This example describes the tags for all your instances.

```
PROMPT> ec2-describe-tags --filter "resource-type=instance"
TAG i-5f4e3d2a instance webserver
TAG i-5f4e3d2a instance stack Production
TAG i-12345678 instance database_server
TAG i-12345678 instance stack Test
```

This example describes the tags for all your instances that have a tag with the key `webserver`.

```
PROMPT> ec2-describe-tags --filter "resource-type=instance" --filter "key=web
server"
TAG i-5f4e3d2a instance webserver
```

This example describes the tags for all your instances that have a tag with the key `stack` and a value of either `Test` or `Production`.

```
PROMPT> ec2-describe-tags --filter "resource-type=instance" --filter "key=stack"
--filter "value=Test" --filter "value=Production"
```

```
TAG i-5f4e3d2a instance stack Production
TAG i-12345678 instance stack Test
```

This example describes the tags for all your instances that have a tag with the key `Purpose` and no value.

```
PROMPT> ec2-describe-tags --filter "resource-type=instance" --filter "key=Purpose" --filter "value="
```

## Changing Tag Values

The following example changes the value of the `stack` tag for one of your AMIs from `Production` to `Test` using the `ec2-create-tags` command.

```
PROMPT> ec2-create-tags ami-1a2b3c4d --tag stack=Test
TAG ami-1a2b3c4d image stack Test
```

## Deleting Tags

The following example deletes the tags that you originally assigned to the AMI and instance.

```
PROMPT> ec2-delete-tags ami-1a2b3c4d i-6f5d4e3a --tag webserver --tag stack
```

Notice that you didn't need to specify a value. If you specify a value, the tag is deleted only if its value matches the one you specified. If you specify an empty string as the value, the tag is deleted only if the tag's value is an empty string. The following example specifies an empty string as the value for the tag to delete (notice the equal sign after `Owner`).

```
PROMPT> ec2-delete-tags snap-1a2b3c4d --tag Owner=
```

## Filtering the List of Resources by Tags

You can describe your resources and filter the results based on the tags. For example, suppose that you've added the tags with the following keys to all your Amazon Elastic Block Store (Amazon EBS) volumes: an `Owner` key and a `Purpose` key. You have a series of teams (TeamA, TeamB, TeamC, and so on), and each team owns a series of volumes. You can get tag descriptions and filter the results by volume.

```
PROMPT> ec2-describe-tags --filter resource-type=volume
TAG vol-abcd1234 volume Owner TeamA
TAG vol-abcd1234 volume Purpose Project1
TAG vol-eefba9876 volume Owner TeamA
TAG vol-eefba9876 volume Purpose Project2
TAG vol-4562dabf volume Owner TeamA
TAG vol-4562dabf volume Purpose RawLogData
TAG vol-2a3d4b5f volume Owner TeamB
TAG vol-2a3d4b5f volume Purpose Project1
TAG vol-9f8g7d6c volume Owner TeamB
TAG vol-9f8g7d6c volume Purpose Project2
TAG vol-3b3a4c4d volume Owner TeamB
TAG vol-3b3a4c4d volume Purpose Logs
TAG vol-1234abcd volume Owner TeamC
```

```
TAG vol-1234abcd volume Purpose Project1
TAG vol-7f7g7d7a volume Owner TeamC
TAG vol-7f7g7d7a volume Purpose Project2
TAG vol-4a4b4c4d volume Owner TeamC
TAG vol-4a4b4c4d volume Purpose Logs
```

Alternatively, you can get volume descriptions and filter the results by tag. The filter name you use is tag:*key*. To get a list of just the volumes belonging to either TeamA or TeamB that contain log data, use **ec2-describe-volumes** with a filter based on the tags of interest. You can use a wildcard to find the volumes that have a tag with the Purpose key that have a value that contains Log.

```
PROMPT> ec2-describe-volumes --filter tag:Owner=TeamA --filter tag:Owner=TeamB
--filter tag:Purpose=*Log*
VOLUME vol-4562dabf 5 us-east-1b available 2010-02-22T22:50:43+0000
Owner TeamA Purpose RawLogData
VOLUME vol-3b3a4c4d 12 us-east-1b available 2010-05-01T13:09:27+0000
Owner TeamB Purpose Logs
```

## Using the API

Use the following API actions to add, update, list, and delete the tags for your resources. The documentation for each API action provides examples.

Description	API Action
Adds or overwrites one or more tags for the specified resource or resources.	<a href="#">CreateTags</a>
Deletes the specified tags from the specified resource or resources.	<a href="#">DeleteTags</a>
Describes one or more tags for your resources.	<a href="#">DescribeTags</a>

In the following sections, you learn how you can use the API to list tags, or list resources and filter the results based on tags.

### Topics

- [Filtering a List of Tags by Resource \(p. 555\)](#)
- [Filtering a List of Resources by Tag \(p. 558\)](#)

## Filtering a List of Tags by Resource

You can describe your tags and filter the results based on the resource type. For example, suppose that you've added tags with the following keys to all your Amazon Elastic Block Store (Amazon EBS) volumes: an Owner key and a Purpose key. You have a series of teams (TeamA, TeamB, TeamC, and so on), and each team owns a series of volumes. The following query gets a list of all tags and filters the results to include only those assigned to a volume.

```
https://ec2.amazonaws.com/?Action=DescribeTags
&Filter.1.Name=resource-type
&Filter.1.Value=volume
&AUTHPARAMS
```

The response includes 18 tags covering 9 volumes.

```
<DescribeTagsResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
 <requestId>7a62c49f-347e-4fc4-9331-6e8eEXAMPLE</requestId>
 <>tagSet>
 <item>
 <resourceId>vol-abcd1234</resourceId>
 <resourceType>volume</resourceType>
 <key>Owner</key>
 <value>TeamA</value>
 </item>
 <item>
 <resourceId>vol-abcd1234</resourceId>
 <resourceType>volume</resourceType>
 <key>Purpose</key>
 <value>Project1</value>
 </item>
 <item>
 <resourceId>vol-efba9876</resourceId>
 <resourceType>volume</resourceType>
 <key>Owner</key>
 <value>TeamA</value>
 </item>
 <item>
 <resourceId>vol-efba9876</resourceId>
 <resourceType>volume</resourceType>
 <key>Purpose</key>
 <value>Project2</value>
 </item>
 <item>
 <resourceId>vol-4562dabf</resourceId>
 <resourceType>volume</resourceType>
 <key>Owner</key>
 <value>TeamA</value>
 </item>
 <item>
 <resourceId>vol-4562dabf</resourceId>
 <resourceType>volume</resourceType>
 <key>Purpose</key>
 <value>RawLogData</value>
 </item>
 <item>
 <resourceId>vol-2a3d4b5f</resourceId>
 <resourceType>volume</resourceType>
 <key>Owner</key>
 <value>TeamB</value>
 </item>
 <item>
 <resourceId>vol-2a3d4b5f</resourceId>
 <resourceType>volume</resourceType>
 <key>Purpose</key>
 <value>Project1</value>
 </item>
 <item>
 <resourceId>vol-9f8g7d6c</resourceId>
 <resourceType>volume</resourceType>
 <key>Owner</key>
 <value>TeamB</value>
 </item>
```

```
</item>
<item>
<resourceId>vol-9f8g7d6c</resourceId>
<resourceType>volume</resourceType>
<key>Purpose</key>
<value>Project2</value>
</item>
<item>
<resourceId>vol-3b3a4c4d</resourceId>
<resourceType>volume</resourceType>
<key>Owner</key>
<value>TeamB</value>
</item>
<item>
<resourceId>vol-3b3a4c4d</resourceId>
<resourceType>volume</resourceType>
<key>Purpose</key>
<value>Logs</value>
</item>
<item>
<resourceId>vol-1234abcd</resourceId>
<resourceType>volume</resourceType>
<key>Owner</key>
<value>TeamC</value>
</item>
<item>
<resourceId>vol-1234abcd</resourceId>
<resourceType>volume</resourceType>
<key>Purpose</key>
<value>Project1</value>
</item>
<item>
<resourceId>vol-7f7g7d7a</resourceId>
<resourceType>volume</resourceType>
<key>Owner</key>
<value>TeamC</value>
</item>
<item>
<resourceId>vol-7f7g7d7a</resourceId>
<resourceType>volume</resourceType>
<key>Purpose</key>
<value>Project2</value>
</item>
<item>
<resourceId>vol-4a4b4c4d</resourceId>
<resourceType>volume</resourceType>
<key>Owner</key>
<value>TeamC</value>
</item>
<item>
<resourceId>vol-4a4b4c4d</resourceId>
<resourceType>volume</resourceType>
<key>Purpose</key>
<value>Logs</value>
</item>
</tagSet>
</DescribeTagsResponse>
```

## Filtering a List of Resources by Tag

You can describe all your resources of a particular resource type and filter the results based on their tags. For example, suppose that you've added tags with the following keys to all your Amazon EBS volumes: an `Owner` key and a `Purpose` key. You have a series of teams (TeamA, TeamB, TeamC, and so on), and each team owns a series of volumes. The following Query gets a list of all volumes and filters the results to include only those volumes with specified tags (volumes belonging to either TeamA or TeamB that contain log data.) You use a wildcard to find the volumes that have a tag with the `Purpose` key that have a value that contains `Log`.

```
https://ec2.amazonaws.com/?Action=DescribeVolumes
&Filter.1.Name=tag:Owner
&Filter.1.Value.1=TeamA
&Filter.1.Value.2=TeamB
&Filter.2.Name=tag:Purpose
&Filter.2.Value.1=*Log*
&AuthParams
```

Because of the filtering, the response includes only two of the volumes that were in the preceding list of tagged volumes. The volume's tags are included in the response.

```
<DescribeVolumesResponse xmlns="http://ec2.amazonaws.com/doc/2013-02-01/">
 <requestId>7a62c49f-347e-4fc4-9331-6e8eEXAMPLE</requestId>
 <volumeSet>
 <item>
 <volumeId>vol-4562dabf</volumeId>
 <size>5</size>
 <snapshotId/>
 <availabilityZone>us-east-1b</availabilityZone>
 <status>available</status>
 <createTime>2010-02-22T22:50:43+0000</createTime>
 <attachmentSet/>
 <tagSet>
 <item>
 <key>Owner</key>
 <value>TeamA</value>
 </item>
 <item>
 <key>Purpose</key>
 <value>RawLogData</value>
 </item>
 </tagSet>
 </item>
 <item>
 <volumeId>vol-3b3a4c4d</volumeId>
 <size>12</size>
 <snapshotId/>
 <availabilityZone>us-east-1b</availabilityZone>
 <status>available</status>
 <createTime>2010-05-01T13:09:27+0000</createTime>
 <attachmentSet/>
 <tagSet>
 <item>
 <key>Owner</key>
 <value>TeamB</value>
 </item>
 </tagSet>
 </item>
 </volumeSet>
</DescribeVolumesResponse>
```

```
<item>
 <key>Purpose</key>
 <value>Logs</value>
</item>
</tagSet>
</item>
</volumeSet>
</DescribeVolumesResponse>
```

# Setting Up the Amazon EC2 Command Line Interface Tools on Linux/UNIX

---

The Amazon EC2 command line interface tools (also called the *CLI tools*) wrap the Amazon EC2 API actions. These tools are written in Java and include shell scripts for both Windows and Linux/UNIX/Mac OSX. For a detailed reference guide that describes the commands, see the [Amazon Elastic Compute Cloud Command Line Reference](#).

Before you can use the Amazon EC2 CLI tools, you need to download them and configure them to use your AWS account. You can set up the tools on your own computer or on an Amazon EC2 instance.

**Note**

For more information about installing the CLI tools on Windows, see [Installing the Amazon EC2 Command Line Interface Tools on Windows](#).

Complete the following tasks to set up an Amazon EC2 environment:

1. [Download the Command Line Interface Tools \(p. 560\)](#)
2. [Get Your Access Key ID and Secret Access Key \(p. 561\)](#)
3. [Set Up the Command Interface Line Tools \(p. 561\)](#)
4. [Download an SSH Client \(p. 565\)](#)

## Download the Command Line Interface Tools

The CLI tools are available as a ZIP file on this site: [Amazon EC2 CLI Tools](#). The ZIP file is self-contained; no installation is required. You can simply download the file and unzip it.

**Important**

Instances that you launch using the Amazon Linux AMI already include the CLI tools.

**Optional step before you unzip:** For detailed information about authenticating the download before unzipping the file, see [Verify the Signature of the Tools Download \(p. 565\)](#).

# 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 use the CLI and make Query requests. 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 Amazon Web Services web site at <http://aws.amazon.com>.
2. Click **My Account/Console**, and then click **Security Credentials**.  
If you are not already logged in, you are prompted to do so.
3. Under the **Access Credentials** section, ensure that the **Access Keys** tab is selected.
4. Locate an active access key under **Your Access Keys**.
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.

# Set Up the Command Interface Line Tools

After you've downloaded the CLI tools, you must complete the following setup tasks before you can use them.

## Topics

- [Tell the Tools Where Java Lives \(p. 561\)](#)
- [Tell the CLI Tools Where They Live \(p. 562\)](#)
- [Tell the CLI Tools Who You Are \(p. 563\)](#)
- [\(Optional\) Tell the CLI Tools To Use a Proxy Server \(p. 563\)](#)
- [Verify the Tools Setup \(p. 564\)](#)
- [\(Optional\) Set the Region \(p. 564\)](#)

## Tell the Tools Where Java Lives

The Amazon EC2 CLI tools require Java. If you don't have Java 1.6 or later installed, download and install Java. Either a JRE or JDK installation is acceptable. To view and download JREs for a range of platforms, see [Java Downloads](#).

### Important

Instances that you launch using the Amazon Linux AMI already include Java.

The Amazon EC2 CLI read the `JAVA_HOME` environment variable to locate the Java runtime. This environment variable should specify the full path of the directory that contains a subdirectory named `bin` that contains the Java executable you installed (`java . exe`).

### To set the `JAVA_HOME` environment variable on Linux/UNIX and Mac OS

1. You can verify whether you have Java installed and where it is located using the following command:

```
$ which java
```

The following is example output.

```
/usr/bin/java
```

Note that this location can be a symlink rather than the actual directory location. You can't use a symlink in an environment variable.

In the default installation of Mac OS, Java is installed in /Library/JAVA/Home. If you installed Java manually, Java might be located elsewhere. For more information about setting this variable on a Mac, see [Unable to find a \\$JAVA\\_HOME](#).

2. Set `JAVA_HOME` to the full path of the Java home directory. For example, if your Java executable is in `/usr/bin`, set `JAVA_HOME` to `/usr`, as shown here.

```
$ export JAVA_HOME=/usr
```

**Note**

The `export` command applies only to the current shell session. To permanently create or update an environment variable, include the command in a start-up script. For example, if you use Bash shell, you can include commands in your `~/.bashrc` or `/etc/profile` file.

**Note**

If you are using Cygwin, `JAVA_HOME` should contain a Windows path.

3. You can verify your `JAVA_HOME` setting using this command.

```
$ $JAVA_HOME/bin/java -version
```

If you've set the environment variable correctly, the output looks something like this.

```
java version "1.7.0_05"
Java(TM) SE Runtime Environment (build 1.7.0_05-b05)
Java HotSpot(TM) Client VM (build 23.1-b03, mixed mode, sharing)
```

## Tell the CLI Tools Where They Live

The CLI tools read the `EC2_HOME` environment variable to locate supporting libraries. Before using these tools, set `EC2_HOME` to the directory path where you unzipped the CLI tools. This directory is named `ec2-api-tools-w.x.y.z` (where `w`, `x`, `y`, and `z` are components of the version number). It contains sub-directories named `bin` and `lib`.

In addition, to make things a little easier, you can add the `bin` directory for the tools to your system path. The examples in the *Amazon Elastic Compute Cloud User Guide* assume that you have done so.

You can set the `EC2_HOME` and `PATH` environment variables as follows. The `export` command updates the current shell session only. To permanently set these environment variables, include these commands in the start-up script for your shell.

### To set the `EC2_HOME` and `PATH` environment variables on Linux/UNIX

1. Use this command to set the `EC2_HOME` environment variable.

```
$ export EC2_HOME=<path-to-tools>
```

**Note**

If you are using Cygwin, `EC2_HOME` must use Linux/UNIX paths (for example, `/usr/bin` instead of `C:\usr\bin`). Additionally, the value of `EC2_HOME` cannot contain any spaces, even if the value is quoted or the spaces are escaped.

2. You can update your `PATH` as follows.

```
$ export PATH=$PATH:$EC2_HOME/bin
```

## Tell the CLI Tools Who You Are

When you sign up for an AWS account, we create access credentials for you so that you can make secure requests to AWS. You must provide these credentials to the Amazon EC2 CLI tools so that they know that the commands that you issue come from your account. For more information, see [Get Your Access Key ID and Secret Access Key \(p. 561\)](#).

You can specify these credentials with the `--aws-access-key` and `--aws-secret-key` (or `-O` and `-W`) options every time you issue a command. However, it's easier to store your access credentials using the following environment variables:

- `AWS_ACCESS_KEY`—Your access key ID
- `AWS_SECRET_KEY`—Your secret access key

If these environment variables are set properly, their values serve as the default values for these required options, so you can omit them from the commands.

**Note**

Although we don't encourage it, for a limited time you can still use `EC2_PRIVATE_KEY` and `EC2_CERT` instead of `AWS_ACCESS_KEY` and `AWS_SECRET_KEY`. For more information, see [Deprecated Options in Common Options for CLI Tools](#) in the *Amazon Elastic Compute Cloud Command Line Reference*. If you specify both sets of credentials, the command line tools use the access key ID and secret access key.

You can set these environment variables as follows. The `export` command updates the current shell session only. To permanently set these environment variables, include these commands in the start-up script for your shell.

```
$ export AWS_ACCESS_KEY=your-aws-access-key
$ export AWS_SECRET_KEY=your-aws-secret-key
```

## (Optional) Tell the CLI Tools To Use a Proxy Server

If the computer you have installed the CLI tools on requires the use of a proxy server, you must tell the CLI tools to use the proxy server with the `EC2_JVM_ARGS` environment variable.

The following table contains the proxy configuration properties that can be set for the `EC2_JVM_ARGS` variable. The properties that are required will depend on the type of proxy server being used. For example, the `http.proxyDomain` and `http.proxyWorkstation` properties are only used with a Windows NTLM proxy.

Property	Description
https.proxyHost	HTTPS proxy host. Use when EC2_URL specifies an HTTPS host.
https.proxyPort	HTTPS proxy port. Use when EC2_URL specifies an HTTPS host.
http.proxyHost	HTTP proxy host. Use when EC2_URL specifies an HTTP host.
http.proxyPort	HTTP proxy port. Use when EC2_URL specifies an HTTP host.
http.proxyDomain	Proxy domain (HTTPS and HTTP)
http.proxyWorkstation	Proxy workstation (HTTPS and HTTP)
http.proxyUser	Proxy user name (HTTPS and HTTP)
http.proxyPass	Proxy password (HTTPS and HTTP)

You set the EC2\_JVM\_ARGS variable with the **export** command:

```
export EC2_JVM_ARGS="-Dhttps.proxyHost=my.proxy.com -Dhttps.proxyPort=8080"
```

## Verify the Tools Setup

Let's quickly verify that your CLI tools are set up correctly. Run the following command to view your available regions.

```
PROMPT> ec2-describe-regions
REGION us-east-1 ec2.us-east-1.amazonaws.com
REGION ap-northeast-1 ec2.ap-northeast-1.amazonaws.com
REGION ap-southeast-1 ec2.ap-southeast-1.amazonaws.com
..
```

The result shows the region names and corresponding service endpoints.

If you get an error that required option **-O** is missing, check the setting of AWS\_ACCESS\_KEY, fix any errors, and try the command again.

If you get an error that required option **-W** is missing, check the setting of AWS\_SECRET\_KEY, fix any errors, and try the command again.

## (Optional) Set the Region

By default, the Amazon EC2 CLI tools use the US East (Northern Virginia) region (`us-east-1`) with the `ec2.us-east-1.amazonaws.com` service endpoint URL. To access a different region with the CLI tools, you must set the EC2\_URL environment variable to the proper service endpoint URL.

### To set the service endpoint URL

1. To list your available service endpoint URLs, call the **ec2-describe-regions** command, as shown in the previous section.

2. Set the EC2\_URL environment variable using the service endpoint URL returned from the **ec2-describe-regions** command as follows.

```
$ export EC2_URL=https://<service_endpoint>
```

If you've already launched an instance using the console and wish to work with the instance using the command line interface, you must specify the endpoint URL for the instance's region. You can verify the region for the instance by checking the region selector in the console navigation bar.

For more information about the regions and endpoints for Amazon EC2, see [Regions and Endpoints](#) in the *Amazon Web Services General Reference*.

## Download an SSH Client

To connect to a Linux instance, you'll need an SSH client. Most Linux and UNIX installations include an SSH client by default. If yours doesn't, the OpenSSH project provides a free implementation of the full suite of SSH tools. For more information, see <http://www.openssh.org>.

Windows users can download and install PuTTY, a free SSH client. To download the client and installation instructions, go to <http://www.chiark.greenend.org.uk/~sgtatham/putty/>. For information about how to use PuTTY with Amazon EC2, see [Connecting to Linux/UNIX Instances from Windows Using PuTTY \(p. 301\)](#).

## Verify the Signature of the Tools Download

Whenever you download an application from the Internet, you should authenticate the identity of the software publisher and check that the application has not been altered or corrupted since it was published. This protects you from installing a version of the application that contains a virus or other malicious code.

If you determine that the software for the CLI tools has been altered, do not unzip or install the file that you downloaded. Instead, contact Amazon Web Services.

### Topics

- [Overview \(p. 565\)](#)
- [Install the GPG Tools \(p. 566\)](#)
- [Authenticate the Public Key \(p. 567\)](#)
- [Verify the Signature of the Package \(p. 568\)](#)

## Overview

The first step is to establish trust with the software publisher: download the public key of the software publisher, check that the owner of the public key is who they claim to be, and then add the public key to your keyring. Your keyring is a collection of known public keys. You can then explicitly trust the public key, or trust it implicitly because the public key is trusted by someone with whom you have a pre-existing trust relationship.

After you've established the authenticity of the public key, you can use it to verify the signature of the application. Using security tools, you'll calculate a signature from the publisher's public key and your downloaded copy of the application. If the calculated signature matches the signature the software

developer has published for the application, you can have confidence that the application has not been altered.

Amazon EC2 command line tools are signed using GnuPG, an open implementation of the Pretty Good Privacy (OpenPGP) standard for secure digital signatures. GnuPG provides authentication and integrity checking through a 128-bit digital signature. Amazon EC2 publishes a public key and signatures you can use to verify the downloaded Amazon EC2 command line tools. For more information about PGP and GnuPG (GPG), see <http://www.gnupg.org>.

## Install the GPG Tools

If your operating system is Linux or UNIX, the GPG tools are likely already installed. To test whether the tools are installed on your system, type `gpg` at a command-line prompt. If the GPG tools are installed, you get a GPG command prompt. If the GPG tools are not installed, you get an error stating that the command cannot be found. You can install the GnuPG package from a repository.

### To install GPG Tools on Debian-based Linux

- From a terminal, run the following command.

```
apt-get install gnupg
```

### To install GPG Tools on Red Hat-based Linux

- From a terminal, run the following command.

```
yum install gnupg
```

### To install GPG Tools on Windows

- Download and install [Gpg4win](#), an implementation of GnuPG that runs on Windows.

During installation of Gpg4win, you can use the default values suggested by Gpg4win. As part of the installation process, you are asked whether you want to define a root certificate. Defining a root certificate is a way to establish trust with many software publishers; setting a root certificate establishes trust with all publishers trusted by that certificate. If you want to define a root certificate, follow the instructions in the text box. If not, click the check box to skip this step. Either option is fine for the purpose of verifying the signature of the Amazon EC2 CLI tools package.

### To install GPG Tools on Mac OS

- Download and install [GPGTools](#), an implementation of GnuPG that runs on Mac OS.

In addition to installing pre-compiled implementations of GnuPG, you can also download and compile the source code from <http://gnupg.org/download/index.en.html>.

After you have installed a set of GPG tools, use them to create a public-private key set. You'll need this later to sign changes to your trust status. If you have previously installed the GPG tools and already have a public-private key set, you can skip this step.

### To create a private key for the GPG tools

1. From the command line, run the following command.

```
gpg --gen-key
```

2. Answer the questions that follow. You can use the suggested default values. Make a note of the passphrase you use to create the private key. You'll need this value later.

## Authenticate the Public Key

The next step in the process is to authenticate the EC2 Packages public key and add it as a trusted key in your GPG keyring.

### To authenticate the EC2 Packages public key

1. Create a text file named `ec2-packages-public.key` and copy the public key from [EC2 Packages Public Key](#) into the text file. This includes everything from `-----BEGIN PGP PUBLIC KEY BLOCK-----` to `-----END PGP PUBLIC KEY BLOCK-----`. Save the text file.

**Note**

This text file must use ASCII encoding.

2. Import the EC2 Packages public key into your keyring using the following command in the directory where you saved the file `ec2-packages-public.key`.

```
gpg --import ec2-packages-public.key
```

The command returns results similar to the following. Make a note of the key value; you'll need it in the next step. In the example below, the key value is `0349E66A`.

```
pg: key 0349E66A: public key "AWS EC2 Packages <ec2-packages@amazon.com>" imported
gpg: Total number processed: 1
gpg: imported: 1 (RSA: 1)
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0 valid: 2 signed: 1 trust: 0-, 0q, 0n, 0m, 0f, 2u
gpg: depth: 1 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 1f, 0u
gpg: next trustdb check due at 2014-07-20
```

3. Verify the fingerprint by running the following command, where `key-value` is replaced by the value from the previous step.

```
gpg --fingerprint key-value
```

This command returns results similar to the following. Compare the key fingerprint to that published on [EC2 Packages Public Key](#). They should match. If they don't, do not continue to install the CLI tools, and contact Amazon Web Services.

```
pub 4096R/0349E66A 2011-04-04
Key fingerprint = A262 37CF 2294 C30E 9844 96C9 116B 3651 0349 E66A
uid AWS EC2 Packages <ec2-packages@amazon.com>
```

4. If the fingerprint matches the one published on the aws.amazon.com website, you may choose to trust EC2 Packages public key. To do so, run the following command, where *key-value* is replaced by the key value from Step 2.

```
gpg --edit-key key-value
```

Type the following command at the GPG tools prompt.

```
trust
```

The GPG tools ask you to establish a level of trust for EC2 Packages, with a prompt such as the following. To learn more about these trust options, go to the [The GNU Privacy Handbook](#).

```
Please decide how far you trust this user to correctly verify other users'
keys
(by looking at passports, checking fingerprints from different sources,
etc.)

1 = I don't know or won't say
2 = I do NOT trust
3 = I trust marginally
4 = I trust fully
5 = I trust ultimately
m = back to the main menu
```

```
Your decision?
```

Type 4 and press the Enter key.

5. Sign the key with your private key (created when you installed the GPG tools) to set the new trust level. Do this using the following command.

```
sign
```

You are asked to confirm, type *y*, and press the Enter key. You are asked for the passcode that you used when you created your private key. Type your passcode and press the Enter key.

6. Save your changes using the following command.

```
save
```

This saves your changes to the keyring. It should also exit the PGP session. If it doesn't, press CTRL+Z to exit the PGP session and return to the main terminal.

## Verify the Signature of the Package

With the GPG tools installed and the EC2 Packages public key authenticated and the EC2 Packages public key trusted, you're ready to check the signature of the Amazon EC2 CLI tools package.

### To verify the signature of Amazon EC2 CLI tools package

1. Download the Amazon EC2 command line interface tools, `ec2-api-tools.zip`, from [Amazon EC2 CLI Tools](#).

2. Create a new text file name `ec2-api-tools.zip.asc` and copy the contents of the Amazon EC2 [command line tools signature](#) into this file. Copy everything including the `-----BEGIN PGP SIGNATURE-----` to `-----END PGP SIGNATURE-----` lines. Save the file.
3. Verify the signature of the CLI tools by typing the following at a command line prompt in the directory where you saved the file `ec2-api-tools.zip.asc` and the command line tool ZIP file `ec2-api-tools.zip`. Both files must be present.

```
gpg --verify ec2-api-tools.zip.asc ec2-api-tools.zip
```

The output should be something like the following.

```
gpg: Signature made Mon Mar 12 14:51:33 2012 PDT using RSA key ID 0349E66A
gpg: Good signature from "AWS EC2 Packages <ec2-packages@amazon.com>"
```

If the output contains the phrase 'Good signature from "AWS EC2 Packages <ec2-packages@amazon.com>"' the signature has successfully been verified, and you can proceed to unzip and install the Amazon EC2 CLI tools. If the output includes the phrase "BAD signature", check that you performed the procedure correctly. If you continue to get this response, contact Amazon Web Services and do not unzip or install the file that you downloaded.

# Making API Requests

---

We provide the Query API for Amazon EC2, as well as software development kits (SDK) for Amazon Web Services (AWS) that enable you to access Amazon EC2 from your preferred programming language.

For more information about the Query API, see [Query Requests \(p. 571\)](#).

## Topics

- [Required Knowledge \(p. 570\)](#)
- [Endpoints \(p. 570\)](#)
- [Available Libraries \(p. 571\)](#)
- [Query Requests \(p. 571\)](#)
- [SOAP Requests \(p. 575\)](#)

## Required Knowledge

If you plan to access Amazon EC2 through an API, you should be familiar with the following:

- XML
- Web services
- HTTP requests
- One or more programming languages, such as Java, PHP, Perl, Python, Ruby, C#, or C++.

## Endpoints

An endpoint is a URL that serves as an entry point for a web service. You can select a regional endpoint for Amazon EC2 when you make your requests to reduce latency. For more information about regions, see [Region and Availability Zone Concepts \(p. 116\)](#). For information about the endpoints for Amazon EC2, see [Regions and Endpoints](#) in the *Amazon Web Services General Reference*.

If you specify the general endpoint, `ec2.amazonaws.com`, we use the endpoint for `us-east-1`. To use a different region, specify its associated endpoint. For example, if you specify `ec2.us-west-2.amazonaws.com` as the endpoint, we direct your request to the `us-west-2` endpoint.

# 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 submitting a request over HTTP or HTTPS. These libraries provide basic functions that automatically take care of tasks such as cryptographically signing your requests, retrying requests, and handling error responses, so that it is easier for you to get started.

For more information, see [AWS SDKs and Tools](#).

# Query Requests

Query requests are HTTP or HTTPS requests that use the HTTP verb GET or POST and a Query parameter named `Action`. For a list of Amazon EC2 API actions, see [Actions](#).

## Topics

- [Structure of a GET Request \(p. 571\)](#)
- [Query Parameters \(p. 572\)](#)
- [Query API Authentication \(p. 572\)](#)
- [Query Response Structures \(p. 573\)](#)
- [Query API Request Rate \(p. 573\)](#)
- [Eventual Consistency \(p. 574\)](#)

## Structure of a GET Request

The Amazon EC2 documentation presents the GET requests as URLs, which can be used directly in a browser.

### Tip

Because the GET requests are URLs, you must URL encode the parameter values. In the Amazon EC2 documentation, we leave the example GET requests unencoded to make them easier to read.

The URL consists of the following:

### Endpoint

The URL that serves as the entry point for the web service

### Action

The action that you want to perform (for example, use `RunInstances` to run an instance)

### Parameters

Any parameters for the request; each parameter is separated by an ampersand (&).

The following is an example request that launches instances.

```
https://ec2.amazonaws.com/?Action=RunInstances&ImageId=ami-60a54009&MaxCount=3&MinCount=1&Placement.AvailabilityZone=us-east-1b&Monitoring.Enabled=true&AWSAccessKeyId=0GS7553JW74RRM612K02EXAMPLE&Version=2013-02-01&Expires=2010-10-10T12:00:00Z&Signature=lBP67vCvGlDMBQ1dofZxg8E8SUEEXAMPLE&SignatureVersion=2&SignatureMethod=HmacSHA256
```

To make these example requests even easier to read, the Amazon EC2 documentation presents them in the following format.

```
https://ec2.amazonaws.com/?Action=RunInstances
&ImageId=ami-2bb65342
&MaxCount=3
&MinCount=1
&Placement.AvailabilityZone=us-east-1b
&Monitoring.Enabled=true
&AWSAccessKeyId=0GS7553JW74RRM612K02EXAMPLE
&Version=2013-02-01
&Expires=2010-10-10T12:00:00Z
&Signature=lBP67vCvG1DMBQ1dofZxg8E8SUEEXAMPLE
&SignatureVersion=2
&SignatureMethod=HmacSHA256
```

The first line specifies the endpoint of the request. After the endpoint is a question mark (?), which separates the endpoint from the parameters.

The Action parameter indicates the action to perform. For a complete list of actions, see [Actions](#) in the *Amazon Elastic Compute Cloud API Reference*.

The remaining lines specify additional parameters for the request.

## Query Parameters

Each Query request must include required common parameters to handle authentication and selection of an action. For more information, see [Common Query Parameters](#) in the *Amazon Elastic Compute Cloud API Reference*.

Some operations take lists of parameters. These lists are specified using the *param.n* notation, where *n* is an integer starting from 1.

The following example adds multiple devices to a block device mapping using a list of BlockDeviceMapping parameters.

```
http://ec2.amazonaws.com/?Action=RunInstances
&ImageId.1=ami-72aa081b
...
&BlockDeviceMapping.1.DeviceName=/dev/sdj
&BlockDeviceMapping.1.Ebs.NoDevice=true
&BlockDeviceMapping.2.DeviceName=/dev/sdh
&BlockDeviceMapping.2.Ebs.VolumeSize=300
&BlockDeviceMapping.3.DeviceName=/dev/sdc
&BlockDeviceMapping.3.VirtualName=ephemeral1
&AUTHPARAMS
```

## 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. Amazon EC2 uses Signature Version 2. For more information, see [Signature Version 2 Signing Process](#) in the *Amazon Web Services General Reference*.

### Tip

In the example Query requests we present in the Amazon EC2 documentation, we omit the parameters related to authentication to make it easier for you to focus on the parameters for the

action. We replace them with the following literal string to remind you that you must include these parameters in your request: `&AuthParams`.

# Query Response Structures

In response to a Query request, the service returns an XML data structure that conforms to an XML schema defined as part of the [WSDL](#) file for Amazon EC2. The structure of an 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 `keySet` for key pairs (see the example that follows). Item elements are children of containers, and their contents vary according to the container's role.

Every response includes a request ID in a `requestId` element. The value is a unique string that AWS assigns. If you ever have issues with a particular request, AWS will ask for the request ID to help troubleshoot the issue. The following shows an example response.

## Query API Request Rate

We throttle Amazon EC2 API requests for each AWS account to help the performance of the service. We ensure that all calls to the Amazon EC2 API (whether they originate from an application, calls to the Amazon EC2 command line interface, or the Amazon EC2 console) don't exceed the maximum allowed API request rate. Note that API requests made by IAM users are attributed to the underlying AWS account.

The Amazon EC2 API actions are divided into the following categories:

- Describe actions, such as `DescribeInstances` and `DescribeVolumes`. These requests simply retrieve cached data, so they have the highest request limit.
  - Modify actions, such as `RunInstances` and `CreateVolumes`. These requests create or modify resources, so they have a lower request limit than describe calls.
  - The `CreateKeyPair`, `GetConsoleOutput`, `AuthorizeSecurityGroupIngress`, and `RevokeSecurityGroupIngress` actions. These requests take the most time and resource to complete, so they have the lowest request limit.

If an API request exceeds the API request rate for its category, the request returns the RequestLimitExceeded error code. To prevent this error, ensure that your application doesn't retry API requests at a high rate. You can do this by using care when polling and by using exponential back-off retries.

Polling

Your application might need to call an API repeatedly to check for an update in status. Before you start polling, give the request time to potentially complete. When you begin polling, use an appropriate sleep interval between successive requests. For best results, use an increasing sleep interval.

### Retries or batch processing

Your application might need to retry an API request after it fails, or to process multiple resources (for example, all your volumes). To lower the rate of API requests, use an appropriate sleep interval between successive requests. For best results, use an increasing or variable sleep interval.

## Eventual Consistency

The Amazon EC2 API follows an eventual consistency model, due to the distributed nature of the system supporting the API. This means that the result of an API command you run that affects your Amazon EC2 resources might not be immediately visible to subsequent commands you run. You should keep this in mind when you carry out an API command that immediately follows a previous API command.

Eventual consistency can affect the way you manage your resources. For example, if you run a command to create a resource, it will eventually be visible to other commands. This means that if you run a command to modify or describe the resource that you just created, its ID might not have propagated throughout the system, and you will get an error responding that the resource does not exist.

To manage eventual consistency, you can do the following:

- Confirm the state of the resource before you run a command to modify it. To do this, run the appropriate `Describe` command.
- Run the appropriate `Describe` command using an exponential back-off algorithm to ensure that you allow enough time for the previous command to propagate through the system. To do this, run the `Describe` command repeatedly, starting with a couple of seconds of wait time, and increasing gradually up to about a minute of wait time. This should ensure that the `Describe` command eventually returns an accurate result.

### Eventual Consistency Error Examples

The following are examples of error codes you may encounter as a result of eventual consistency.

- `InvalidInstanceID.NotFound`

If you successfully carry out the `RunInstances` command, and run a command directly afterwards using the instance ID that was provided in the response of `RunInstances`, it may return an `InvalidInstanceID.NotFound` error. This does not mean the instance does not exist.

Some specific commands that may be affected are:

- `DescribeInstances`: To confirm the actual state of the instance, run this command using an exponential back-off algorithm.
- `TerminateInstances`: To confirm the state of the instance, first run the `DescribeInstances` command using an exponential back-off algorithm.

#### Important

If you get an `InvalidInstanceID.NotFound` error after running `TerminateInstances`, this does not mean that the instance is or will be terminated. Your instance could still be running. This is why it is important to first confirm the instance's state using `DescribeInstances`.

- `InvalidGroup.NotFound`

If you successfully carry out the `CreateSecurityGroup` command, and run a command directly afterwards using the instance ID that was provided in the response of `CreateSecurityGroup`, it may return an `InvalidGroup.NotFound` error. To confirm the state of the security group, run the `DescribeSecurityGroups` command using an exponential back-off algorithm.

# SOAP Requests

We have deprecated the SOAP API for Amazon EC2, and will end support for it in 2013.

# Document History

The following table describes important additions to the Amazon EC2 documentation set. We also update the documentation frequently to address the feedback that you send us.

**Current API version: 2013-02-01.**

Feature	API Version	Description	Release Date
Incremental Snapshot Copies	2013-02-01	Added information about incremental snapshot copies. For more information, see <a href="#">Copying an Amazon EBS Snapshot (p. 494)</a> .	11 June 2013
New Tags pane	2013-02-01	Added information about the new Tags pane in the Amazon EC2 console. For more information, see <a href="#">Tagging Your Amazon EC2 Resources (p. 540)</a> .	04 April 2013
New Amazon Linux AMI release	2013-02-01	Added information about the release of Amazon Linux AMI 2013.03. For more information, see <a href="#">Amazon Linux AMIs (p. 71)</a> .	27 March 2013
Support for additional EBS-optimized instance types	2013-02-01	The following instance types can now be launched as EBS-optimized instances: High-CPU Extra Large (c1.xlarge), High-Memory Double Extra Large (m2.2xlarge), M3 Extra Large (m3.xlarge), and M3 Double Extra Large (m3.2xlarge).  For more information, see <a href="#">EBS-Optimized Instances (p. 91)</a> .	19 March 2013
Copy an AMI from one region to another	2013-02-01	You can copy an AMI from one region to another, enabling you to launch consistent instances in more than one AWS region quickly and easily.  For more information, see <a href="#">Copying AMIs (p. 67)</a> .	11 March 2013

Feature	API Version	Description	Release Date
Launch instances into a default VPC	2013-02-01	<p>Your AWS account is capable of launching instances into either the EC2-Classic or EC2-VPC platform, or only into the EC2-VPC platform, on a region-by-region basis. If you can launch instances only into EC2-VPC, we create a default VPC for you. When you launch an instance, we launch it into your default VPC, unless you create a nondefault VPC and specify it when you launch the instance.</p> <p>For more information, see <a href="#">Supported Platforms (p. 120)</a>.</p>	11 March 2013
High-memory cluster (cr1.8xlarge) instance type	2012-12-01	Have large amounts of memory coupled with high CPU and network performance. These instances are well suited for in-memory analytics, graph analysis, and scientific computing applications. For more information, see <a href="#">Instance Families and Types (p. 87)</a> .	21 January 2013
High storage (hs1.8xlarge) instance type	2012-12-01	High storage instances provide very high storage density and high sequential read and write performance per instance. They are well-suited for data warehousing, Hadoop/MapReduce, and parallel file systems. For more information, see <a href="#">High Storage Instances (p. 102)</a> .	20 December 2012
EBS snapshot copy	2012-12-01	You can use snapshot copies to create backups of data, to create new Amazon EBS volumes, or to create Amazon Machine Images (AMIs). For more information, see <a href="#">Copying an Amazon EBS Snapshot (p. 494)</a> .	17 December 2012
Updated EBS metrics and status checks for Provisioned IOPS volumes	2012-10-01	Updated the EBS metrics to include two new metrics for Provisioned IOPS volumes. For more information, see <a href="#">Monitoring Volumes with CloudWatch (p. 478)</a> . Also added new status checks for Provisioned IOPS volumes. For more information, see <a href="#">Monitoring Volumes with Status Checks (p. 480)</a> .	20 November 2012

Feature	API Version	Description	Release Date
Support for Microsoft Windows Server 2012	2012-10-01	<p>Amazon EC2 now provides you with several pre-configured Windows Server 2012 AMIs. These AMIs are immediately available for use in every Region and for every 64-bit instance type. The AMIs support the following languages:</p> <ul style="list-style-type: none"> <li>• English</li> <li>• Chinese Simplified</li> <li>• Chinese Traditional</li> <li>• Chinese Traditional Hong Kong</li> <li>• Japanese</li> <li>• Korean</li> <li>• Portuguese</li> <li>• Portuguese Brazil</li> <li>• Czech</li> <li>• Dutch</li> <li>• French</li> <li>• German</li> <li>• Hungarian</li> <li>• Italian</li> <li>• Polish</li> <li>• Russian</li> <li>• Spanish</li> <li>• Swedish</li> <li>• Turkish</li> </ul> <p>For more information, see <a href="#">Windows Instance Types (p. 92)</a>.</p>	19 November 2012
Linux Kernels	2012-10-01	Updated AKI IDs; reorganized distribution kernels; updated PVOps section.	13 November 2012
Amazon Linux AMIs	2012-10-01	Added information to Adding Packages and the location of the latest Linux AMIs. For more information, see <a href="#">Adding Packages (p. 75)</a> .	02 November 2012
M3 Instances	2012-10-01	Added information about the new Amazon Elastic Compute Cloud (Amazon EC2) M3 extra-large and M3 double-extra-large instance types. For more information, see <a href="#">Instance Families and Types (p. 87)</a> .	31 October 2012
Amazon EC2 Spot Instance Request Status	2012-10-01	Added information about Amazon EC2 Spot Instance request status, which makes it easy to determine the state of your Amazon EC2 Spot requests.	14 October 2012

Feature	API Version	Description	Release Date
New Amazon Linux AMI release	2012-08-15	Added information about the release of Amazon Linux AMI 2012.09. For more information, see <a href="#">Amazon Linux AMIs (p. 71)</a> .	11 October 2012
Amazon EC2 Reserved Instance Marketplace	2012-08-15	The Reserved Instance Marketplace matches sellers who have Amazon EC2 Reserved Instances that they no longer need with buyers who are looking to purchase additional capacity. Reserved Instances bought and sold through the Reserved Instance Marketplace work like any other Reserved Instances, except that they can have less than a full standard term remaining and can be sold at different prices.	11 September 2012
Provisioned IOPS (input/output operations per second) for Amazon EBS	2012-07-20	Provisioned IOPS volumes deliver predictable, high performance for I/O intensive workloads, such as database applications, that rely on consistent and fast response times. For more information, see <a href="#">EBS Volume Types (p. 458)</a> .	31 July 2012
High I/O instances for Amazon EC2	2012-06-15	High I/O instances provides very high, low latency, disk I/O performance using SSD-based local instance storage. For more information, see <a href="#">High I/O Instances (p. 101)</a> .	18 July 2012
IAM roles on Amazon EC2 instances	2012-06-01	IAM roles for Amazon EC2 provide: <ul style="list-style-type: none"> <li>• AWS access keys for applications running on Amazon EC2 instances.</li> <li>• Automatic rotation of the AWS access keys on the Amazon EC2 instance.</li> <li>• Granular permissions for applications running on Amazon EC2 instances that make requests to your AWS services.</li> </ul>	11 June 2012
Spot Instance features that make it easier to get started and handle the potential of interruption.		Using Auto Scaling, you can now manage your Spot Instances: <ul style="list-style-type: none"> <li>• Place bids for Amazon EC2 Spot Instances using Auto Scaling launch configurations, and set up a schedule for placing bids for Spot Instances. For more information, see <a href="#">Managing Spot Instances with Auto Scaling (p. 161)</a>.</li> <li>• Get notifications when instances are launched or terminated. For more information, see <a href="#">Using Auto Scaling to Get Notifications for Spot Instances (p. 174)</a>.</li> <li>• Use AWS CloudFormation templates to launch Spot Instances in a stack with AWS resources. For more information, see <a href="#">Using CloudFormation Templates to Launch Spot Instances (p. 176)</a>.</li> </ul>	7 June 2012

Feature	API Version	Description	Release Date
EC2 instance export and timestamps for status checks for Amazon EC2	2012-05-01	Added support for exporting Windows Server instances that you originally imported into EC2. Added support for timestamps on instance status and system status to indicate the date and time that a status check failed.	25 May 2012
EC2 instance export, and timestamps in instance and system status checks for Amazon VPC	2012-05-01	Added support for EC2 instance export to Citrix Xen, Microsoft Hyper-V, and VMware vSphere. Added support for timestamps in instance and system status checks.	25 May 2012
Cluster Compute Eight Extra Large instances	2012-04-01	Added support for cc2.8xlarge instances in Amazon Virtual Private Cloud (Amazon VPC).	26 April 2012
AWS Marketplace AMIs	2012-04-01	Added support for AWS Marketplace AMIs.	19 April 2012
New Linux AMI release		Added information about the release of Amazon Linux AMI 2012.03. For more information, see <a href="#">Amazon Linux AMIs (p. 71)</a> .	28 March 2012
New AKI version		Added information about the release of new AKI version 1.03 and the release of AKIs for the AWS GovCloud (US) region. For more information, see <a href="#">Enabling Your Own Linux Kernels (p. 77)</a> .	28 March 2012
Medium instances, support for 64-bit on all AMIs, and a Java-based SSH Client		Added support for a new instance type and 64-bit information. Added procedures for using the Java-based SSH client to connect to Linux/UNIX instances.	7 March 2012
Reserved Instance pricing tiers		Added a new section discussing how to take advantage of the discount pricing that is built into the Reserved Instance pricing tiers. For more information, see <a href="#">Understanding Discount Pricing Tiers (p. 229)</a> .	5 March 2012
Elastic Network Interfaces (ENIs) for EC2 instances in Amazon Virtual Private Cloud		Added new section about elastic network interfaces (ENIs) for EC2 instances in a VPC. For more information, see <a href="#">Elastic Network Interfaces (ENI) (p. 442)</a> .	21 December 2011
New GRU Region and AKIs		Added information about the release of new AKIs for the SA-East-1 Region. This release deprecates the AKI version 1.01. AKI version 1.02 will continue to be backward compatible.	14 December 2011
New offering types for Amazon EC2 Reserved Instances	2011-11-01	You can choose from a variety of Reserved Instance offerings that address your projected use of the instance: <i>Heavy Utilization</i> , <i>Medium Utilization</i> , and <i>Light Utilization</i> . See <a href="#">Reserved Instances (p. 212)</a> .	01 December 2011

Feature	API Version	Description	Release Date
Amazon EC2 instance status		You can view additional details about the status of your instances, including scheduled events planned by AWS that might have an impact on your instances. These operational activities include instance reboots required to apply software updates or security patches, or instance retirements required where there are hardware issues. See <a href="#">Monitoring the Status of Your Instances (p. 354)</a> .	16 November 2011
Amazon EC2 Cluster Compute Instance Type		Added support for Cluster Compute Eight Extra Large (cc2.8xlarge) to Amazon EC2.	14 November 2011
New PDX Region and AKIs		Added information about the release of new AKI's for the new US-West 2 Region.	8 November 2011
Amazon EC2 Spot Instances in Amazon VPC		Added information about the support for Amazon EC2 Spot Instances in Amazon VPC. With this update, users will be able to launch Spot Instances in the Amazon Virtual Private Cloud (Amazon VPC). By launching Spot Instances in Amazon VPC, users of Spot Instances can enjoy all of the controls and advanced security options of Amazon VPC. For more information, see <a href="#">Launching Spot Instances in Amazon Virtual Private Cloud (p. 178)</a> .	11 October 2011
New Linux AMI release		Added information about the release of Amazon Linux AMI 2011.09. This update removes the beta tag from the Amazon Linux AMI, supports the ability to lock the repositories to a specific version, and provides for notification when updates are available to installed packages including security updates. For more information, see <a href="#">Amazon Linux AMIs (p. 71)</a> .	26 September 2011
Simplified VM import process for users of the CLI tools		The VM Import process for CLI users is simplified with the enhanced functionality of <code>ec2-import-instance</code> and <code>ec2-import-volume</code> , which now will perform the upload of the images into Amazon EC2 after creating the import task. In addition, with the introduction of the <code>ec2-resume-import</code> command, users can restart an incomplete upload at the point the task stopped. For more information, see <a href="#">Importing Your Virtual Machine into Amazon EC2 (p. 330)</a> .	15 September 2011

Feature	API Version	Description	Release Date
Support for importing in VHD file format		VM Import can now import virtual machine image files in VHD format. The VHD file format is compatible with the Citrix Xen and Microsoft Hyper-V virtualization platforms. With this release, VM Import now supports RAW, VHD and VMDK (VMware ESX-compatible) image formats. For more information, see <a href="#">Using the Command Line Tools to Import Your Virtual Machine to Amazon EC2 (p. 324)</a> .	24 August 2011
Support for Microsoft Windows Server 2003 R2		VM Import now supports Windows Server 2003 (R2). With this release, VM Import supports all versions of Microsoft Windows Server supported by Amazon EC2.	24 August 2011
Update to the Amazon EC2 VM Import Connector for VMware vCenter		Added information about the 1.1 version of the Amazon EC2 VM Import Connector for VMware vCenter virtual appliance (Connector). This update includes proxy support for Internet access, better error handling, improved task progress bar accuracy, and several bug fixes. For more information, see <a href="#">Importing EC2 Instances (p. 308)</a> .	27 June 2011
Enabling Linux AMI to run user-provided kernels		Added information about the AKI version change from 1.01 to 1.02. This version updates the PVGRUB to address launch failures associated with t1.micro Linux instances. For more information, go to <a href="#">Enabling Your Own Linux Kernels (p. 77)</a> .	20 June 2011
Spot Instances Availability Zone pricing changes		Added information about the Spot Instances Availability Zone pricing feature. In this release, we've added new Availability Zone pricing options as part of the information returned when you query for Spot Instance requests and Spot Price history. These additions make it easier to determine the price required to launch a Spot Instance into a particular Availability Zone. For more information, see <a href="#">Spot Instances (p. 132)</a> .	26 May 2011
AWS Identity and Access Management		Added information about AWS Identity and Access Management (IAM), which enables users to specify which Amazon EC2 actions a user can use with Amazon EC2 resources in general. For more information, see <a href="#">Controlling Access to Amazon EC2 Resources (p. 421)</a> .	26 April 2011
Enabling Linux AMI to run user-provided kernels		Added information about enabling a Linux AMI to use PVGRUB Amazon Kernel Image (AKI) to run a user-provided kernel. For more information, go to <a href="#">Enabling Your Own Linux Kernels (p. 77)</a> .	26 April 2011

Feature	API Version	Description	Release Date
Dedicated instances		Launched within your Amazon Virtual Private Cloud (Amazon VPC), Dedicated Instances are instances that are physically isolated at the host hardware level. Dedicated Instances let you take advantage of Amazon VPC and the AWS cloud, with benefits including on-demand elastic provisioning and pay only for what you use, while isolating your Amazon EC2 compute instances at the hardware level. For more information, see <a href="#">Using EC2 Dedicated Instances</a> in the <i>Amazon Virtual Private Cloud User Guide</i> .	27 March 2011
Reserved Instances updates to the AWS Management Console		Updates to the AWS Management Console make it easier for users to view their Reserved Instances and purchase additional Reserved Instances, including Dedicated Reserved Instances. For more information, see <a href="#">Reserved Instances (p. 212)</a> .	27 March 2011
Support for Windows Server 2008 R2		Amazon EC2 now provides you with several pre-configured Windows Server 2008 R2 AMIs. These AMIs are immediately available for use in every Region and in most 64-bit instance types, excluding t1.micro and HPC families. The AMIs will support multiple languages. For more information, see <a href="#">Windows Instance Types (p. 92)</a> .	15 March 2011
New Amazon Linux reference AMI		Added information about the new Amazon Linux reference AMI, which replaces the CentOS reference AMI. Removed information about the CentOS reference AMI, including the section named Correcting Clock Drift for Cluster Instances on CentOS 5.4 AMI. For more information, see <a href="#">Reference AMIs (p. 106)</a> .	15 March 2011
Metadata information		Added information about metadata to reflect changes in the 2011-01-01 release. For more information, see <a href="#">Instance Metadata and User Data (p. 365)</a> and <a href="#">Instance Metadata Categories (p. 371)</a> .	11 March 2011
Amazon EC2 VM Import Connector for VMware vCenter		Added information about the Amazon EC2 VM Import Connector for VMware vCenter virtual appliance (Connector). The Connector is a plug-in for VMware vCenter that integrates with VMware vSphere Client and provides a graphical user interface that you can use to import your VMware virtual machines to Amazon EC2. For more information, see <a href="#">Importing EC2 Instances (p. 308)</a> .	3 March 2011
Force volume detachment		You can now use the AWS Management Console to force the detachment of an Amazon EBS volume from an instance. For more information, see <a href="#">Detaching an Amazon EBS Volume from an Instance (p. 489)</a> .	23 February 2011

Feature	API Version	Description	Release Date
Instance termination protection		You can now use the AWS Management Console to prevent an instance from being terminated. For more information, see <a href="#">Enabling Termination Protection for an Instance (p. 381)</a> .	23 February 2011
Correcting Clock Drift for Cluster Instances on CentOS 5.4 AMI		Added information about how to correct clock drift for cluster instances running on Amazon's CentOS 5.4 AMI.	25 January 2011
Restructured Documentation		<p>Implemented the following updates:</p> <ul style="list-style-type: none"> <li>• Consolidated all developer guide information into this user guide.</li> <li>• Restructured and updated the information about creating AMIs. For more information, see <a href="#">Creating Your Own AMIs (p. 30)</a>.</li> <li>• Updated the introduction to the user guide. For more information, see <a href="#">What is Amazon EC2? (p. 1)</a></li> </ul>	14 January 2011
VM Import		Added information about VM Import, which allows you to import a virtual machine or volume into Amazon EC2. For more information, see <a href="#">Using the Command Line Tools to Import Your Virtual Machine to Amazon EC2 (p. 324)</a> .	15 December 2010
Basic monitoring for instances		Added information about basic monitoring for EC2 instances. For more information, see <a href="#">Monitoring Instances (p. 343)</a> .	12 December 2010
Cluster GPU instances		Amazon EC2 offers cluster GPU instances (cg1.4xlarge) for high-performance computing (HPC) applications. For more information, see <a href="#">Overview (p. 104)</a> .	14 November 2010
Filters and Tags		Added information about listing, filtering, and tagging resources. For more information, see <a href="#">Listing and Filtering Your Resources (p. 535)</a> and <a href="#">Tagging Your Amazon EC2 Resources (p. 540)</a> .	19 September 2010
Idempotent Instance Launch		Added information about ensuring idempotency when running instances. For more information, see <a href="#">Ensuring Idempotency (p. 363)</a> .	19 September 2010
Micro instances		Amazon EC2 offers the t1.micro instance type for certain types of applications. For more information, see <a href="#">Micro Instances (p. 94)</a> .	8 September 2010
AWS Identity and Access Management for Amazon EC2		Amazon EC2 now integrates with AWS Identity and Access Management (IAM). For more information, see <a href="#">Controlling Access to Amazon EC2 Resources (p. 421)</a> .	2 September 2010

Feature	API Version	Description	Release Date
Cluster instances		Amazon EC2 offers cluster compute instances for high-performance computing (HPC) applications. For more information, see <a href="#">Overview (p. 104)</a> .	12 July 2010
Amazon VPC IP Address Designation		Amazon VPC users can now specify the IP address to assign an instance launched in a VPC.	12 July 2010
Amazon CloudWatch Monitoring for Amazon EBS Volumes		Amazon CloudWatch monitoring is now automatically available for Amazon EBS volumes. For more information, see <a href="#">Monitoring Volumes with CloudWatch (p. 478)</a> .	14 June 2010
High-memory extra large instances		Amazon EC2 now supports a High-Memory Extra Large (m2.xlarge) instance type. For more information, see <a href="#">Available Instance Types (p. 88)</a> .	22 February 2010
Reserved Instances with Windows		Amazon EC2 now supports Reserved Instances with Windows. For more information about Reserved Instances, see <a href="#">Reserved Instances (p. 212)</a> .	22 February 2010