Amazon Elastic Compute Cloud

User Guide for Linux

API Version 2014-10-01

Amazon Elastic Compute Cloud: User Guide for Linux

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

Table of Contents

What Is Amazon EC2?

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

For more information about cloud computing, see What is Cloud Computing?

Features of Amazon EC2

Amazon EC2 provides the following features:

- Virtual computing environments, known as *instances*
- Preconfigured templates for your instances, known as *Amazon Machine Images (AMIs)*, that package the bits you need for your server (including the operating system and additional software)
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as *instance types*
- Secure login information for your instances using *key pairs* (AWS stores the public key, and you store the private key in a secure place)
- Storage volumes for temporary data that's deleted when you stop or terminate your instance, known as *instance store volumes*
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as *Amazon EBS volumes*
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as *regions* and *Availability Zones*
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using *security groups*
- Static IP addresses for dynamic cloud computing, known as *Elastic IP addresses*
- Metadata, known as *tags*, that you can create and assign to your Amazon EC2 resources
- Virtual networks you can create that are logically isolated from the rest of the AWS cloud, and that you can optionally connect to your own network, known as *virtual private clouds* (VPCs)

For more information about the features of Amazon EC2, see the Amazon EC2 product page.

For more information about running your website on AWS, see Websites & Website Hosting.

How to Get Started with Amazon EC2

The first thing you need to do is get set up to use Amazon EC2. After you are set up, you are ready to complete the Getting Started tutorial for Amazon EC2. Whenever you need more information about a feature of Amazon EC2, you can read the technical documentation.

Get Up and Running

Basics

Networking and Security

Storage

Working with Linux Instances

If you have questions about whether AWS is right for you, contact AWS Sales. If you have technical questions about Amazon EC2, use the Amazon EC2 forum.

Related Services

You can provision Amazon EC2 resources, such as instances and volumes, directly using Amazon EC2. You can also provision Amazon EC2 resources using other services in AWS. For more information, see the following documentation:

- Auto Scaling Developer Guide
- AWS CloudFormation User Guide
- AWS Elastic Beanstalk Developer Guide
- AWS OpsWorks User Guide

To automatically distribute incoming application traffic across multiple instances, use Elastic Load Balancing. For more information, see Elastic Load Balancing Developer Guide.

To monitor basic statistics for your instances and Amazon EBS volumes, use Amazon CloudWatch. For more information, see the Amazon CloudWatch Developer Guide.

To monitor the calls made to the Amazon EC2 API for your account, including calls made by the AWS Management Console, command line tools, and other services, use AWS CloudTrail. For more information, see the AWS CloudTrail User Guide.

To get a managed relational database in the cloud, use Amazon Relational Database Service (Amazon RDS) to launch a database instance. Although you can set up a database on an EC2 instance, Amazon RDS offers the advantage of handling your database management tasks, such as patching the software, backing up, and storing the backups. For more information, see Amazon Relational Database Service Developer Guide.

Accessing Amazon EC2

Amazon EC2 provides a web-based user interface, the Amazon EC2 console. If you've signed up for an AWS account, you can access the Amazon EC2 console by signing into the AWS Management Console and selecting **EC2** from the console home page.

If you prefer to use a command line interface, you have several options:

AWS Command Line Interface (CLI)
Provides commands for a broad set of AWS products, and is supported on Windows, Mac, and Linux. To get started, see AWS Command Line Interface User Guide. For more information about the commands for Amazon EC2, see ec2 in the *AWS Command Line Interface Reference*.

Amazon EC2 Command Line Interface (CLI) Tools
Provides commands for Amazon EC2, Amazon EBS, and Amazon VPC, and is supported on Windows, Mac, and Linux. To get started, see Setting Up the Amazon EC2 Command Line Interface Tools on Linux and Commands (CLI Tools) in the *Amazon EC2 Command Line Reference*.

AWS Tools for Windows PowerShell
Provides commands for a broad set of AWS products for those who script in the PowerShell environment. To get started, see the AWS Tools for Windows PowerShell User Guide. For more information about the cmdlets for Amazon EC2, see the AWS Tools for Windows PowerShell Reference.

Amazon EC2 provides a Query API. These requests are HTTP or HTTPS requests that use the HTTP verbs GET or POST and a Query parameter named `Action`. For more information about the API actions for Amazon EC2, see Actions in the *Amazon EC2 API Reference*.

If you prefer to build applications using language-specific APIs instead of submitting a request over HTTP or HTTPS, AWS provides libraries, sample code, tutorials, and other resources for software developers. These libraries provide basic functions that automate tasks such as cryptographically signing your requests, retrying requests, and handling error responses, making it is easier for you to get started. For more information, see AWS SDKs and Tools.

Pricing for Amazon EC2

When you sign up for AWS, you can get started with Amazon EC2 for free using the AWS Free Tier.

Amazon EC2 provides the following purchasing options for instances:

On-Demand Instances
    Pay for the instances that you use by the hour, with no long-term commitments or up-front payments.

Reserved Instances
    Make a low, one-time, up-front payment for an instance, reserve it for a one- or three-year term, and pay a significantly lower hourly rate for these instances.

Spot Instances
    Specify the maximum hourly price that you are willing to pay to run a particular instance type. The Spot Price fluctuates based on supply and demand, but you never pay more than the maximum price you specified. If the Spot Price moves higher than your maximum price, Amazon EC2 shuts down your Spot Instances.

For a complete list of charges and specific prices for Amazon EC2, see Amazon EC2 Pricing.

To calculate the cost of a sample provisioned environment, see AWS Economics Center.

To see your bill, go to your AWS Account Activity page. Your bill contains links to usage reports that provide details about your bill. To learn more about AWS account billing, see AWS Account Billing.

If you have questions concerning AWS billing, accounts, and events, contact AWS Support.

For an overview of Trusted Advisor, a service that helps you optimize the costs, security, and performance of your AWS environment, see AWS Trusted Advisor.

Instances and AMIs

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 an *instance*, which is a copy of the AMI running as a virtual server in the cloud. You can launch multiple instances of an AMI, as shown in the following figure.
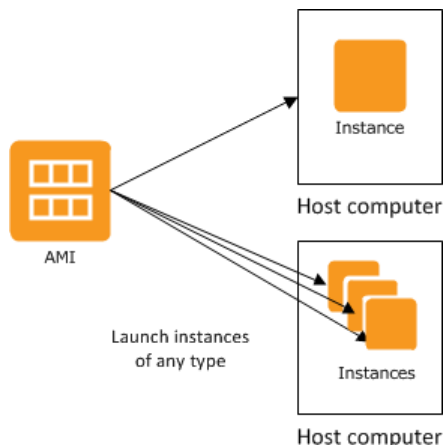
Your instances keep running until you stop or terminate them, or until they fail. If an instance fails, you can launch a new one from the AMI.

Instances

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 about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details.

After you launch an instance, it looks like a traditional host, and you can interact with it as you would any computer. You have complete control of your instances; you can use **sudo** to run commands that require root privileges.

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

In addition to the limit on running instances, there is a 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 two times your running instance limit.

Storage for Your Instance

The root device for your instance contains the image used to boot the instance. For more information, see Amazon EC2 Root Device Volume (p. 14).

Your instance may include local storage volumes, known as instance store volumes, which you can configure at launch time with block device mapping. For more information, see Block Device Mapping (p. 591). After these volumes have been added to and mapped on your instance, they are available for you to mount and use. If your instance fails, or if your instance is stopped or terminated, the data on these volumes is lost; therefore, these volumes are best used for temporary data. For important data, you should use a replication strategy across multiple instances in order to keep your data safe, or store your persistent data in Amazon S3 or Amazon EBS volumes. For more information, see Storage (p. 518).

Security Best Practices

- Use AWS Identity and Access Management (IAM) to control access to your AWS resources, including your instances. You can create IAM users and groups under your AWS account, assign security credentials to each, and control the access that each has to resources and services in AWS. For more information, see Controlling Access to Amazon EC2 Resources (p. 429).
- 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, see Amazon EC2 Security Groups (p. 421).
- 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.
- 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. 67). For more information about sharing AMIs safely, see Shared AMIs (p. 60).

Stopping, Starting, and Terminating Instances

Stopping an instance

When an instance is stopped, the instance performs a normal shutdown, and then transitions to a `stopped` state. All of its Amazon EBS volumes remain attached, and you can start the instance 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 the rate for the new instance type after the instance is started. All of the associated Amazon EBS usage of your instance, including root device usage, is billed using typical Amazon EBS prices.

When an instance is in a stopped state, you can attach or detach Amazon EBS volumes. You can also create an AMI from the instance, and you can change the kernel, RAM disk, and instance type.

Terminating an instance

When an instance is terminated, the instance performs a normal shutdown, then the attached Amazon EBS volumes are deleted unless the volume's `deleteOnTermination` attribute is set to `false`. The instance itself is also deleted, and you can't start the instance again at a later time.

To prevent accidental termination, you can disable instance termination. If you do so, ensure that the `disableApiTermination` attribute is set to `true` for the instance. To control the behavior of an 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 volumes for the root device default to `stop`, and instances with instance-store root devices are always terminated as the result of an instance shutdown.

For more information, see Instance Lifecycle (p. 286).

AMIs

Amazon Web Services (AWS) publishes many Amazon Machine Images (AMIs) that contain common software configurations for public use. In addition, members of the AWS developer community have published their own custom AMIs. You can also create your own custom AMI or AMIs; doing so enables you to quickly and easily start new instances that have everything you need. For example, if your application is a website or a 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.

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. 55).

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.

> **Note**
> Some AWS resources might not be available in all regions and Availability Zones. Ensure that you can create the resources you need in the desired regions or Availability Zone before deploying your applications.

Topics

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.
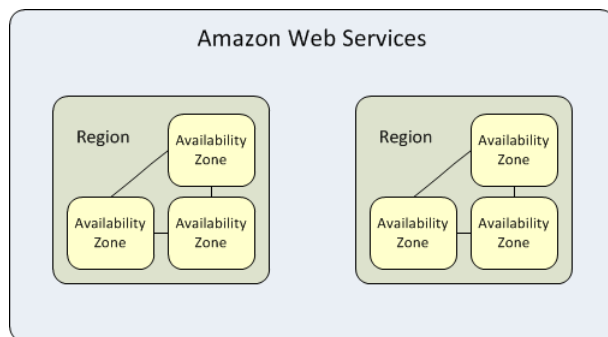
Amazon EC2 resources are either global, tied to a region, or tied to an Availability Zone. For more information, see Resource Locations (p. 605).

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. The following table lists the regions that provide support for Amazon EC2.

Code	Name
ap-northeast-1	Asia Pacific (Tokyo)
ap-southeast-1	Asia Pacific (Singapore)
ap-southeast-2	Asia Pacific (Sydney)
eu-central-1	EU (Frankfurt)
eu-west-1	EU (Ireland)
sa-east-1	South America (Sao Paulo)
us-east-1	US East (N. Virginia)
us-west-1	US West (N. California)
us-west-2	US West (Oregon)

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*. For more information about endpoints and protocols in AWS GovCloud (US), see AWS GovCloud (US) Endpoints in the *AWS GovCloud (US) User Guide*.

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 an AMI (p. 85).

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. 9).

When you launch an instance, you can select an Availability Zone or let us choose 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. 489).

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 command line interface to determine which regions and Availability Zones are available for your use. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

To find your regions and Availability Zones using the AWS Management Console

1. Open the AWS Management Console.
2. From the navigation bar, view the options in the region selector.

```
                          Oregon ▲

      US East (N. Virginia)

    | US West (Oregon)

      US West (N. California)

      EU (Ireland)

      EU (Frankfurt)

      Asia Pacific (Singapore)

      Asia Pacific (Tokyo)

      Asia Pacific (Sydney)

      South America (São Paulo)
```
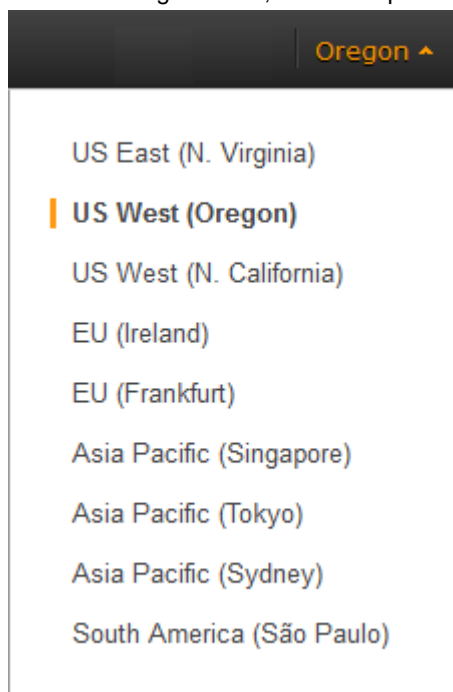
3. Open the Amazon EC2 console. Your Availability Zones are listed on the dashboard under **Service Health**, under **Availability Zone Status**.

Service Health

Service Status:

✅ US East (N. Virginia):
This service is operating normally

Availability Zone Status:

✅ us-east-1b:
Availability zone is operating normally

✅ us-east-1c:
Availability zone is operating normally

✅ us-east-1d:
Availability zone is operating normally

To find your regions and Availability Zones using the AWS CLI

1. Use the describe-regions command as follows to describe your regions.

```
$ aws ec2 describe-regions
{
    "Regions": [
        {
            "Endpoint": "ec2.us-east-1.amazonaws.com",
            "RegionName": "us-east-1"
        },
        {
            "Endpoint": "ec2.ap-southeast-1.amazonaws.com",
            "RegionName": "ap-southeast-1"
        },
        {
            "Endpoint": "ec2.ap-southeast-2.amazonaws.com",
            "RegionName": "ap-southeast-2"
        },
        ...
    ]
}
```

2. Use the describe-availability-zones command as follows to describe your Availability Zones within the us-east-1 region.

```
$ aws ec2 describe-availability-zones --region us-east-1
{
    "AvailabilityZones": [
        {
            "State": "available",
            "RegionName": "us-east-1",
            "Messages": [],
            "ZoneName": "us-east-1b"
        },
```

```
        {
            "State": "available",
            "RegionName": "us-east-1",
            "Messages": [],
            "ZoneName": "us-east-1c"
        },
        {
            "State": "available",
            "RegionName": "us-east-1",
            "Messages": [],
            "ZoneName": "us-east-1d"
        }
    ]
}
```

To find your regions and Availability Zones using the Amazon EC2 CLI

1. Use the ec2-describe-regions command as follows 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
..
```

2. Use the ec2-describe-availability-zones command as follows to describe your 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. You can specify the region for a resource using the AWS Management Console or the command line.

To specify the region for a resource using the console

1. Open the Amazon EC2 console.
2. Use the region selector in the navigation bar.
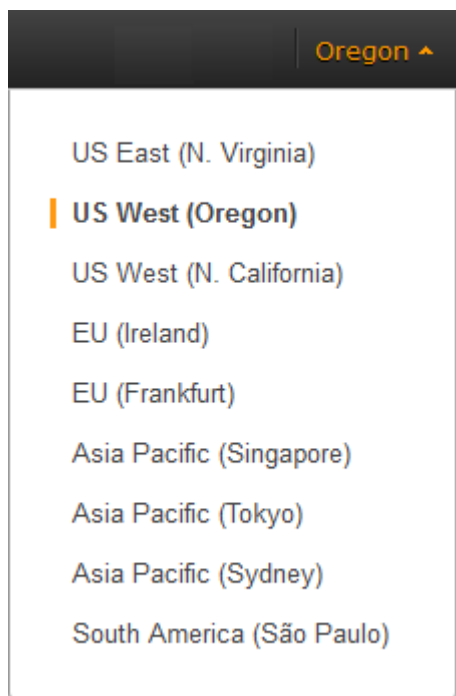
To specify the default region using the command line

You can set the value of an environment variable to the desired regional endpoint (for example, `ht-tps://ec2.us-west-1.amazonaws.com`):

- `AWS_DEFAULT_REGION` (AWS CLI)
- `EC2_URL` (Amazon EC2 CLI)

Alternatively, you can use the `--region` command line option with each individual command. For example, `--region us-west-1`.

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.

To specify an Availability Zone for your instance using the console

1. Open the Amazon EC2 console.
2. On the dashboard, click **Launch Instance**.

3. Follow the directions for the wizard. On the **Configure Instance Details** page, do the following:

    *   [EC2-Classic] Select one of the Availability Zone options from the list, or select **No Preference** to enable us to select the best one for you.

        Availability Zone ⓘ   No preference ▾

    *   [EC2-VPC] Select one of the subnet options from the list, or select **No preference (default subnet in any Availability Zone)** to enable us to select the best one for you.

        Subnet ⓘ   No preference (default subnet in any Availability Zone) ▾   Create new subn

To specify an Availability Zone for your instance using the AWS CLI

You can use the run-instances command with one of the following options:

*   [EC2-Classic] `--placement`
*   [EC2-VPC] `--subnet-id`

To specify an Availability Zone for your instance using the Amazon EC2 CLI

You can use the ec2-run-instances command with one of the following options:

*   [EC2-Classic] `--availability-zone`
*   [EC2-VPC] `--subnet`

Migrating an Instance to Another Availability Zone

If you need to, you can migrate an instance from one Availability Zone to another. For example, if you are trying to modify the instance type of your instance and we can't launch an instance of the new instance type in the current Availability Zone, you could migrate the instance to an Availability Zone where we can launch an instance of that instance type.

The migration process involves creating an AMI from the original instance, launching an instance in the new Availability Zone, and updating the configuration of the new instance, as shown in the following procedure.

To migrate an instance to another Availability Zone

1. Create an AMI from the instance. The procedure depends on the operating system and the type of root device volume for the instance. For more information, see the documentation that corresponds to your operating system and root device volume:

    *   Creating an Amazon EBS-Backed Linux AMI (p. 74)
    *   Creating an Instance Store-Backed Linux AMI (p. 77)
    *   Creating an Amazon EBS-Backed Windows AMI
    *   Creating an Instance Store-Backed Windows AMI

2. [EC2-VPC] If you need to preserve the private IP address of the instance, you must delete the subnet in the current Availability Zone and then create a subnet in the new Availability Zone with the same IP address range as the original subnet. Note that you must terminate all instances in a subnet before you can delete it. Therefore, you should move all instances in the current subnet to the new subnet.

3. Launch an instance from the AMI that you just created, specifying the new Availability Zone or subnet. You can use the same instance type as the original instance, or select a new instance type. For more information, see Launching Instances in an Availability Zone (p. 12).

4. If the original instance has an associated Elastic IP address, associate it with the new instance. For more information, see Associating an Elastic IP Address with a Different Running Instance (p. 492).

5. If the original instance is a Reserved Instance, change the Availability Zone for your reservation. (If you also changed the instance type, you can also change the instance type for your reservation.) For more information, see Submitting Modification Requests (p. 224).

6. (Optional) Terminate the original instance. For more information, see Terminating an Instance (p. 315).

Amazon EC2 Root Device Volume

When you launch an 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 backed 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. 14)
- Choosing an AMI by Root Device Type (p. 16)
- Determining the Root Device Type of Your Instance (p. 17)
- Changing the Root Device Volume to Persist (p. 17)

Root Device Storage Concepts

You can launch an instance from one of two types of AMIs: an instance store-backed AMI or an Amazon EBS-backed AMI. The description of an AMI includes which 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 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. 55).

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, but this data is deleted when the instance is terminated (instance store-backed instances do not support the **Stop** action) or if it fails (such as if an underlying drive has issues).
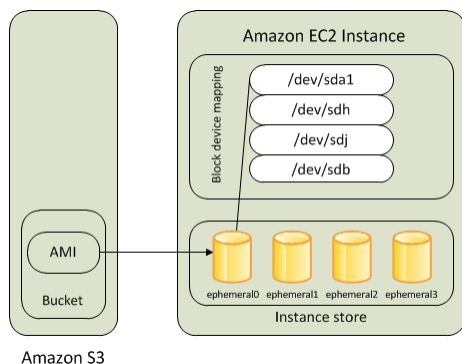
Amazon S3

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 persistent storage on a regular basis.

For more information, see Amazon EC2 Instance Store (p. 581).

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 Amazon EBS snapshot referenced by the AMI you use. You can optionally use other Amazon EBS volumes or instance store volumes.
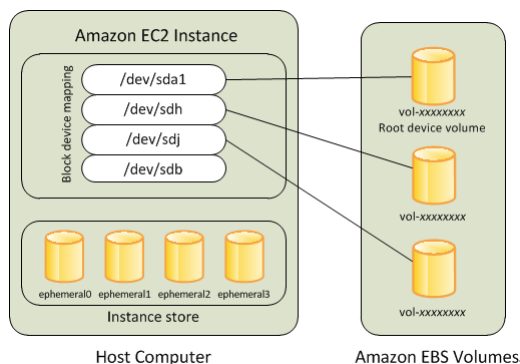
An Amazon EBS-backed instance can be stopped and later restarted without affecting data stored in the attached volumes. There are various instance– and volume-related tasks you can do when an Amazon EBS-backed instance is in a stopped state. For example, you can modify the properties of the instance, 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. 17).

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 an Amazon EBS-Backed Linux AMI (p. 74).
- 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.

Choosing an AMI by Root Device Type

The AMI that you specify when you launch your instance determines the type of root device volume that your instance has.

To choose an Amazon EBS-backed AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. From the filter lists, select the image type (such as **Public images**), the operating system (such as **Amazon Linux**), and **EBS images**.
4. (Optional) To get additional information to help you make your choice, click the **Show/Hide Columns** icon, update the columns to display, and click **Close**.
5. Choose an AMI and write down its AMI ID.

To choose an instance store-backed AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. From the filter lists, select the image type (such as **Public images**), the operating system (such as **Amazon Linux**), and **Instance store images**.
4. (Optional) To get additional information to help you make your choice, click the **Show/Hide Columns** icon, update the columns to display, and click **Close**.
5. Choose an AMI and write down its AMI ID.

To verify the type of the root device volume of an AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-images (AWS CLI)
- ec2-describe-images (Amazon EC2 CLI)
- Get-EC2Image (AWS Tools for Windows PowerShell)

Determining the Root Device Type of Your Instance

To determine the root device type of an instance using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**, and select the instance.
3. Check the value of **Root device type** in the **Description** tab as follows:

   - If the value is `ebs`, this is an Amazon EBS-backed instance.
   - If the value is `instance store`, this is an instance store-backed instance.

To determine the root device type of an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-instances (AWS CLI)
- ec2-describe-instances (Amazon EC2 CLI)
- Get-EC2Instance (AWS Tools for Windows PowerShell)

Changing the Root Device Volume to Persist

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` attribute to `false` using a block device mapping.

Changing the Root Volume to Persist Using the Console

Using the console, you can change the `DeleteOnTermination` attribute when you launch an instance. To change this attribute for a running instance, you must use the command line.

To change the root device volume of an instance to persist at launch using the console

1. Open the Amazon EC2 console.
2. From the Amazon EC2 console dashboard, click **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, choose the AMI to use and click **Select**.
4. Follow the wizard to complete the **Choose an Instance Type** and **Configure Instance Details** pages.
5. On the **Add Storage** page, deselect the **Delete On Termination** check box for the root volume.
6. Complete the remaining wizard pages, and then click **Launch**.

You can verify the setting by viewing details for the root device volume on the instance's details pane. Next to **Block devices**, click the entry for the root device volume. By default, **Delete on termination** is `True`. If you change the default behavior, **Delete on termination** is `False`.

Changing the Root Volume of an Instance to Persist Using the AWS CLI

Using the AWS CLI, you can change the `DeleteOnTermination` attribute when you launch an instance or while the instance is running. The root device is typically `/dev/sda1` (Linux) or `xvda` (Windows).

Example at Launch

Use the run-instances command to preserve the root volume by including a block device mapping that sets its `DeleteOnTermination` attribute for to `false`.

```
$ aws ec2 run-instances --image-id ami-1a2b3c4d --block-device-mappings
'[{"DeviceName":"/dev/sda1","Ebs":{"DeleteOnTermination":false}}]" other para
meters...
```

You can confirm that `DeleteOnTermination` is `false` by using the describe-instances command and looking for the `BlockDeviceMappings` entry for `/dev/sda1` in the command output, as shown here.

```
...
"BlockDeviceMappings": [
    {
        "DeviceName": "/dev/sda1",
        "Ebs": {
            "Status": "attached",
            "DeleteOnTermination": false,
            "VolumeId": "vol-877166c8",
            "AttachTime": "2013-07-19T02:42:39.000Z"
        }
    }
]
...
```

Example While the Instance is Running

Use the modify-instance-attribute command to preserve the root volume by including a block device mapping that sets its `DeleteOnTermination` attribute to `false`.

```
$ aws ec2 modify-instance-attribute --instance-id i-5203422c --block-device-
mappings  '[{"DeviceName":"/dev/sda1","Ebs":{"DeleteOnTermination":false}}]'
```

Changing the Root Volume of an Instance to Persist Using the Amazon EC2 CLI

Using the Amazon EC2 CLI, you can change the `DeleteOnTermination` attribute when you launch an instance or while the instance is running. The root device is typically `/dev/sda1` (Linux) or `xvda` (Windows).

Example at Launch

Use the ec2-run-instances command to 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.

```
$ ec2-run-instances ami-1a2b3c4d  -b "/dev/sda1=::false" other parameters...
-v
```

By running the command in verbose mode, you can see the underlying request and response, and confirm that `DeleteOnTermination` is `false`, as shown here.

```
...
<blockDeviceMapping>
    <item>
        <deviceName>/dev/sda1</deviceName>
        <ebs>
            <deleteOnTermination>false</deleteOnTermination>
        </ebs>
    </item>
</blockDeviceMapping>
...
```

Example While the Instance is Running

Use the ec2-modify-instance-attribute command to preserve the root volume by setting its `DeleteOnTermination` attribute to `false`.

```
$ ec2-modify-instance-attribute i-5203422c -b "/dev/sda1=::false"
```

Setting Up with Amazon EC2

If you've already signed up for Amazon Web Services (AWS), you can start using Amazon EC2 immediately. You can open the Amazon EC2 console, click **Launch Instance**, and follow the steps in the launch wizard to launch your first instance.

If you haven't signed up for AWS yet, or if you need assistance launching your first instance, complete the following tasks to get set up to use Amazon EC2:

1. Sign Up for AWS (p. 20)
2. Create an IAM User (p. 21)
3. Create a Key Pair (p. 22)
4. Create a Virtual Private Cloud (VPC) (p. 24)
5. Create a Security Group (p. 24)

Sign Up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all services in AWS, including Amazon EC2. You are charged only for the services that you use.

With Amazon EC2, you pay only for what you use. If you are a new AWS customer, you can get started with Amazon EC2 for free. For more information, see AWS Free Tier.

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

To create an AWS account

1. Open http://aws.amazon.com, and then click **Sign Up**.
2. Follow the on-screen instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Note your AWS account number, because you'll need it for the next task.

Create an IAM User

Services in AWS, such as Amazon EC2, require that you provide credentials when you access them, so that the service can determine whether you have permission to access its resources. The console requires your password. You can create access keys for your AWS account to access the command line interface or API. However, we don't recommend that you access AWS using the credentials for your AWS account; we recommend that you use AWS Identity and Access Management (IAM) instead. Create an IAM user, and then add the user to an IAM group with administrative permissions or and grant this user administrative permissions. You can then access AWS using a special URL and the credentials for the IAM user.

If you signed up for AWS but have not created an IAM user for yourself, you can create one using the IAM console.

To create the Administrators group

1. Sign in to the AWS Management Console and open the IAM console at https://con-sole.aws.amazon.com/iam/.
2. In the navigation pane, click **Groups** and then click **Create New Group**.
3. In the **Group Name** box, type `Administrators` and then click **Next Step**.
4. In the **Select Policy Template** section, click **Select** next to the **Administrator Access** policy template.
5. Click **Next Step** and then click **Create Group**.

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

To create the IAM user, add the user to the Administrators group, and create a password for the user

1. In the navigation pane, click **Users** and then click **Create New Users**.
2. In box **1**, type a user name and then click **Create**.
3. Click **Download Credentials** and save your access key in a secure place. You will need your access key for programmatic access to AWS using the AWS CLI, the AWS SDKs, or the HTTP APIs.

   Note
   You cannot retrieve the secret access key after you complete this step; if you misplace it you must create a new one.

   After you have downloaded your access key, click **Close**.
4. In the content pane, under **User Name**, click the name of the user you just created. (You might need to scroll down to find the user in the list.)
5. In the content pane, in the **Groups** section, click **Add User to Groups**.
6. Select the **Administrators** group and then click **Add to Groups**.
7. In the content pane, in the **Security Credentials** section (you might need to scroll down to find this section), under **Sign-In Credentials**, click **Manage Password**.
8. Select **Assign a custom password** and then type and confirm a password. When you are finished, click **Apply**.

To sign in as this new IAM user, sign out of the AWS console, then use the following URL, where *your_aws_account_id* is your AWS account number without the hyphens (for example, if your AWS account number is `1234-5678-9012`, your AWS account ID is `123456789012`):

```
https://your_aws_account_id.signin.aws.amazon.com/console/
```

Enter the IAM user name and password that you just created. When you're signed in, the navigation bar displays "*your_user_name @ your_aws_account_id*".

If you don't want the URL for your sign-in page to contain your AWS account ID, you can create an account alias. From the IAM dashboard, click **Create Account Alias** and enter an alias, such as your company name. To sign in after you create an account alias, use the following URL:

```
https://your_account_alias.signin.aws.amazon.com/console/
```

To verify the sign-in link for IAM users for your account, open the IAM console and check under **IAM users sign-in link** on the dashboard.

For more information about IAM, see IAM and Amazon EC2 (p. 430).

Create a Key Pair

AWS uses public-key cryptography to secure the login information for your instance. A Linux instance has no password; you use a key pair to log in to your instance securely. You specify the name of the key pair when you launch your instance, then provide the private key when you log in using SSH.

If you haven't created a key pair already, you can create one using the Amazon EC2 console. Note that if you plan to launch instances in multiple regions, you'll need to create a key pair in each region. For more information about regions, see Regions and Availability Zones (p. 7).

To create a key pair

1. Open the Amazon EC2 console.
2. From the navigation bar, select a region for the key pair. You can select any region that's available to you, regardless of your location. However, key pairs are specific to a region; for example, if you plan to launch an instance in the US West (Oregon) region, you must create a key pair for the instance in the US West (Oregon) region.
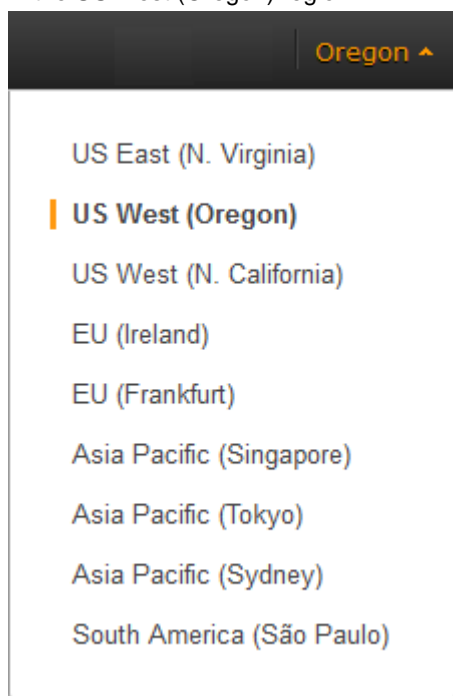
3. Click **Key Pairs** in the navigation pane.

4. Click **Create Key Pair**.

5. Enter a name for the new key pair in the **Key pair name** field of the **Create Key Pair** dialog box, and then click **Create**. Choose a name that is easy for you to remember, such as your IAM user name, followed by `-key-pair`, plus the region name. For example, *me*-key-pair-*uswest2*.

6. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is `.pem`. Save the private key file in a safe place.

    > **Important**
    >
    > This is the only chance for you to save the private key file. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

7. If you will use an SSH client on a Mac or Linux computer to connect to your Linux instance, use the following command to set the permissions of your private key file so that only you can read it.

```
$ chmod 400 your_user_name-key-pair-region_name.pem
```

For more information, see Amazon EC2 Key Pairs (p. 413).

To connect to your instance using your key pair

To your Linux instance from a computer running Mac or Linux, you'll specify the `.pem` file to your SSH client with the `-i` option and the path to your private key. To connect to your Linux instance from a computer running Windows, you can use either MindTerm or PuTTY. If you plan to use PuTTY, you'll need to install it and use the following procedure to convert the `.pem` file to a `.ppk` file.

(Optional) To prepare to connect to a Linux instance from Windows using PuTTY

1. Download and install PuTTY from http://www.chiark.greenend.org.uk/~sgtatham/putty/. Be sure to install the entire suite.

2. Start PuTTYgen (for example, from the **Start** menu, click **All Programs > PuTTY > PuTTYgen**).

3. Under **Type of key to generate**, select **SSH-2 RSA**.
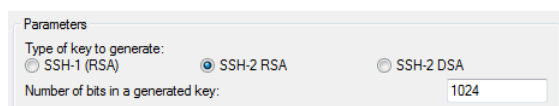
4. Click **Load**. By default, PuTTYgen displays only files with the extension `.ppk`. To locate your `.pem` file, select the option to display files of all types.
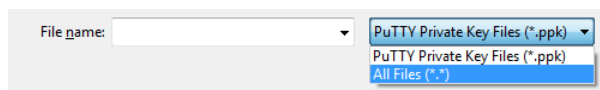
5. Select the private key file that you created in the previous procedure and click **Open**. Click **OK** to dismiss the confirmation dialog box.

6. Click **Save private key**. PuTTYgen displays a warning about saving the key without a passphrase. Click **Yes**.

7. Specify the same name for the key that you used for the key pair. PuTTY automatically adds the `.ppk` file extension.

Create a Virtual Private Cloud (VPC)

Amazon VPC enables you to launch AWS resources into a virtual network that you've defined. If you have a default VPC, you can skip this section and move to the next task, Create a Security Group (p. 24). To determine whether you have a default VPC, see Supported Platforms in the Amazon EC2 Console (p. 471). Otherwise, you can create a nondefault VPC in your account using the steps below.

> **Important**
> If your account supports EC2-Classic in a region, then you do not have a default VPC in that region. T2 instances must be launched into a VPC.

To create a nondefault VPC

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. From the navigation bar, select a region for the VPC. VPCs are specific to a region, so you should select the same region in which you created your key pair.
3. On the VPC dashboard, click **Start VPC Wizard**.
4. On the **Step 1: Select a VPC Configuration** page, ensure that **VPC with a Single Public Subnet** is selected, and click **Select**.
5. On the **Step 2: VPC with a Single Public Subnet** page, enter a friendly name for your VPC in the **VPC name** field. Leave the other default configuration settings, and click **Create VPC**. On the confirmation page, click **OK**.

For more information about Amazon VPC, see What is Amazon VPC? in the *Amazon VPC User Guide*.

Create a Security Group

Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level. You must add rules to a security group that enable you to connect to your instance from your IP address using SSH. You can also add rules that allow inbound and outbound HTTP and HTTPS access from anywhere.

Note that if you plan to launch instances in multiple regions, you'll need to create a security group in each region. For more information about regions, see Regions and Availability Zones (p. 7).

> **Tip**
> You'll need the public IP address of your local computer, which you can get using a service. For example, we provide the following service: http://checkip.amazonaws.com/. To locate another service that provides your IP address, use the search phrase "what is my IP address." If you are connecting through an Internet service provider (ISP) or from behind a firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

To create a security group with least privilege

1. Open the Amazon EC2 console.
2. From the navigation bar, select a region for the security group. Security groups are specific to a region, so you should select the same region in which you created your key pair.
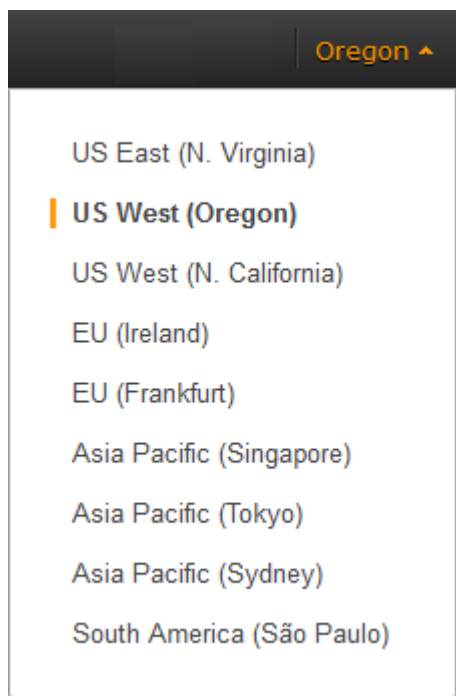
3. Click **Security Groups** in the navigation pane.

4. Click **Create Security Group**.

5. Enter a name for the new security group and a description. Choose a name that is easy for you to remember, such as your IAM user name, followed by _SG_, plus the region name. For example, *me_SG_uswest2*.

6. In the **VPC** list, ensure that your default VPC is selected; it's marked with an asterisk (*).

   Note
   If your account supports EC2-Classic, select the VPC that you created in the previous task.

7. On the **Inbound** tab, create the following rules (click **Add Rule** for each new rule), and then click **Create**:

   - Select **HTTP** from the **Type** list, and make sure that **Source** is set to **Anywhere** (`0.0.0.0/0`).
   - Select **HTTPS** from the **Type** list, and make sure that **Source** is set to **Anywhere** (`0.0.0.0/0`).
   - Select **SSH** from the **Type** list. In the **Source** box, ensure **Custom IP** is selected, and specify the public IP address of your computer or network in CIDR notation. To specify an individual IP address in CIDR notation, add the routing prefix `/32`. For example, if your IP address is `203.0.113.25`, specify `203.0.113.25/32`. If your company allocates addresses from a range, specify the entire range, such as `203.0.113.0/24`.

     Caution
     For security reasons, we don't recommend that you allow SSH access from all IP addresses (`0.0.0.0/0`) to your instance, except for testing purposes and only for a short time.

For more information, see Amazon EC2 Security Groups (p. 421).

Getting Started with Amazon EC2 Linux Instances

Let's get started with Amazon Elastic Compute Cloud (Amazon EC2) by launching, connecting to, and using a Linux instance. We'll use the AWS Management Console, a point-and-click web-based interface, to launch and connect to a Linux instance.

Important
Before you begin, be sure that you've completed the steps in Setting Up with Amazon EC2.

The instance is an Amazon EBS-backed instance (meaning that the root volume is an Amazon EBS volume). We'll also create and attach an additional Amazon EBS volume. You can either specify the Availability Zone in which to launch your instance, or let us select an Availability Zone for you. When you launch your instance, you secure it by specifying a key pair and security group. (You created these when getting set up.) When you connect to your instance, you must specify the private key of the key pair that you specified when launching your instance.
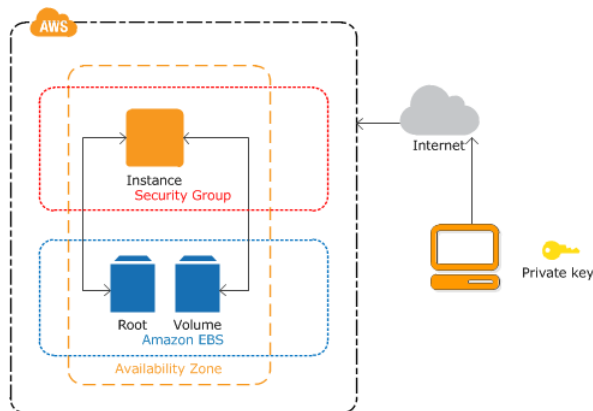
To complete this exercise, perform the following tasks:

1. Launch an Amazon EC2 Instance (p. 27)
2. Connect to Your Instance (p. 28)
3. (Optional) Add a Volume to Your Instance (p. 31)
4. Clean Up Your Instance and Volume (p. 34)

Related Topics

- If you'd prefer to launch a Windows instance, see this tutorial in the *Amazon EC2 User Guide for Microsoft Windows Instances*: Getting Started with Amazon EC2 Windows Instances.
- If you'd prefer to use the AWS CLI, see this tutorial in the *AWS Command Line Interface User Guide*: Using Amazon EC2 through the AWS CLI.
- If you'd prefer to use the Amazon EC2 CLI, see this tutorial in the *Amazon EC2 Command Line Reference*: Launching an Instance Using the Amazon EC2 CLI.

Launch an Amazon EC2 Instance

You can launch a Linux instance using the AWS Management Console as described in this topic. Before you begin, be sure that you've completed the steps in Get Set Up for Amazon EC2. After you've launched your instance, you can connect to it and use it. For more information, see Connect to Your Instance (p. 28).

If you'd prefer to launch and connect to a Windows instance, see Getting Started with Amazon EC2 Windows Instances.

Important
When you sign up for AWS, you can get started with Amazon EC2 for free using the AWS Free Tier. If you created your AWS account less than 12 months ago, and have not already exceeded the free tier benefits for Amazon EC2 and Amazon EBS, it will not cost you anything to complete this tutorial, because we help you select options that are within the free tier benefits. Otherwise, you'll incur the standard Amazon EC2 usage fees from the time that you launch the instance until you terminate the instance (which is the final task of this tutorial), even if it remains idle. The total charges to complete this tutorial are minimal (typically only a few dollars).

The following procedure is intended to help you launch your first instance quickly and doesn't go through all possible options. For more information about the advanced options, see Launching an Instance.

To launch an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the console dashboard, click **Launch Instance**.
3. The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations called Amazon Machine Images (AMIs) that serve as templates for your instance. Select the 64-bit Amazon Linux AMI. Notice that this configuration is marked "Free tier eligible."
4. On the **Choose an Instance Type** page, you can select the hardware configuration of your instance. The `t2.micro` instance type is selected by default. Alternatively, select **All generations** from the filter list, and then select the `t1.micro` instance type. Note that these are the only instance types eligible for the free tier.

    Note
    T2 instances must be launched into a VPC. If your AWS account supports EC2-Classic and you do not have any VPCs, the launch wizard creates a VPC for you. Otherwise, if you have one or more VPCs, click **Next: Configure Instance Details** to select a VPC and subnet.

5. Click **Review and Launch** to let the wizard complete the other configuration settings for you.
6. On the **Review Instance Launch** page, under **Security Groups**, you'll see that the wizard created and selected a security group for you. Instead, select the security group that you created when getting set up using the following steps:

    a. Click **Edit security groups**.

    b. On the **Configure Security Group** page, ensure the **Select an existing security group** option is selected.

    c. Select your security group from the list of existing security groups, and click **Review and Launch**.

7. On the **Review Instance Launch** page, click **Launch**.

8. In the **Select an existing key pair or create a new key pair** dialog box, select **Choose an existing key pair**, then select the key pair you created when getting set up.

   Alternatively, you can create a new key pair. Select **Create a new key pair**, enter a name for the key pair, and then click **Download Key Pair**. This is the only chance for you to save the private key file, so be sure to download it. Save the private key file in a safe place. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

   A key pair enables you to connect to a Linux instance through SSH. Therefore, don't select the **Proceed without a key pair** option. If you launch your instance without a key pair, then you can't connect to it.

   When you are ready, select the acknowledgment check box, and then click **Launch Instances**.

9. A confirmation page lets you know that your instance is launching. Click **View Instances** to close the confirmation page and return to the console.

10. On the **Instances** screen, you can view the status of your instance. It takes a short time for an instance to launch. When you launch an instance, its initial state is `pending`. After the instance starts, its state changes to `running`, and it receives a public DNS name. (If the **Public DNS** column is hidden, click the **Show/Hide** icon and select **Public DNS**.)

Connect to Your Instance

After you launch your instance, you can connect to it and use it the way that you'd use a computer sitting in front of you.

> **Note**
> It may take a couple of minutes after launch for your running instance to finish provisioning so that you can connect to it. Check that your instance has passed its status checks - you can view this information in the **Status Checks** column on the **Instances** page. If you receive an error while attempting to connect to your instance, see Troubleshooting Connecting to Your Instance.

Before you try to connect to your instance, be sure that you've completed the following tasks:

- **Get the public DNS name of the instance**
  You can get the public DNS for your instance using the Amazon EC2 console (check the **Public DNS** column; if this column is hidden, click the **Show/Hide** icon and select **Public DNS**). If you prefer, you can use the describe-instances (AWS CLI) or ec2-describe-instances (Amazon EC2 CLI) command.

- **Locate the private key**
  You'll need the fully-qualified path of the `.pem` file for the key pair that you specified when you launched the instance.

- **Enable inbound SSH traffic from your IP address to your instance**
  Ensure that the security group associated with your instance allows incoming SSH traffic from your IP address. For more information, see Authorizing Network Access to Your Instances.

  > **Important**
  > Your default security group does not allow incoming SSH traffic by default.

There are several ways to connect to a Linux instance. Choose the method that meets your needs:

- Option 1: Connect Using Your Browser (p. 29)
- Option 2: Connect from Windows Using PuTTY (p. 30)
- Option 3: Connect from Mac or Linux Using an SSH Client (p. 31)

Next Step

After you've successfully launched and connected to your instance, you can do any of the following:

- Continue to the next step in this tutorial, Add a Volume to Your Instance (p. 31).
- Continue using this instance with a different tutorial, such as Installing a LAMP Web Server or Hosting a WordPress Blog.
- Skip to the last step in this tutorial, Clean Up Your Instance and Volume (p. 34), to terminate the instance so that you don't continue to incur charges.

Option 1: Connect Using Your Browser

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 the following pages: Install Java and Enable Java in your web browser.

To connect to your Linux instance using a web browser

1. From the Amazon EC2 console, click **Instances** in the navigation pane.
2. Select the instance, and then click **Connect**.
3. Click **A Java SSH client directly from my browser (Java required)**.
4. Amazon EC2 automatically detects the public DNS name of your instance and populates **Public DNS** for you. It also detects the key pair that you specified when you launched the instance. Complete the following, and then click **Launch SSH Client**.

   a. In **User name**, enter `ec2-user`.

      Tip
      For Amazon Linux, the user name is `ec2-user`. For RHEL5, the user name is either `root` or `ec2-user`. For Ubuntu, the user name is `ubuntu`. For Fedora, the user name is either `fedora` or `ec2-user`. For SUSE Linux, the user name is `root`. Otherwise, if `ec2-user` and `root` don't work, check with your AMI provider.

   b. In **Private key path**, enter the fully qualified path to your private key (`.pem`) file, including the key pair name; for example:

      `C:\KeyPairs\my-key-pair.pem`

   c. (Optional) Click **Store in browser cache** to store the location of the private key in your browser cache. This enables Amazon EC2 to detect the location of the private key in subsequent browser sessions, until you clear your browser's cache.

5. If necessary, click **Yes** to trust the certificate, and click **Run** to run the MindTerm client.
6. If this is your first time running MindTerm, a series of dialog boxes asks you to accept the license agreement, to confirm setup for your home directory, and to confirm setup of the known hosts directory. Confirm these settings.
7. A dialog prompts you to add the host to your set of known hosts. If you do not want to store the host key information on your local computer, click **No**.
8. A window opens and you are connected to your instance.

   Note
   If you clicked **No** in the previous step, you'll see the following message, which is expected:

   ```
   Verification of server key disabled in this session.
   ```

Option 2: Connect from Windows Using PuTTY

PuTTY doesn't use `.pem` files, it uses `.ppk` files. If you haven't already generated a `.ppk` file, do so now. For more information, see To prepare to connect to a Linux instance from Windows using PuTTY.

To connect to your Linux instance using PuTTY

1. Start PuTTY (from the **Start** menu, click **All Programs > PuTTY > PuTTY**).

2. In the Category pane, select **Session** and complete the following fields:

    a. In the **Host Name** box, enter ec2-user@*public_dns_name*.

    b. Under **Connection type**, select **SSH**.
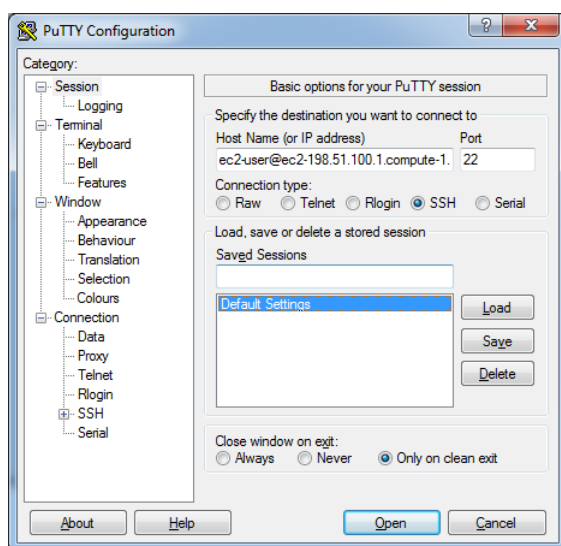
    c. Ensure that **Port** is 22.

3. In the **Category** pane, expand **Connection**, expand **SSH**, and then select **Auth**. Complete the following:

    a. Click **Browse**.

    b. Select the `.ppk` file that you generated for your key pair, and then click **Open**.

    c. Click **Open** to start the PuTTY session.
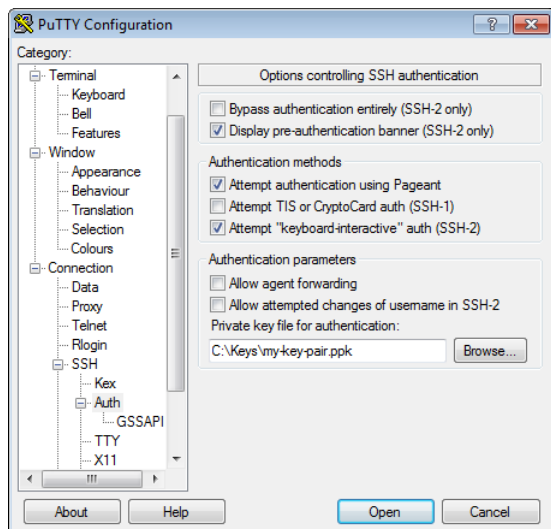
4. If this is the first time you have connected to this instance, PuTTY displays a security alert dialog box that asks whether you trust the host you are connecting to. Click **Yes**. A window opens and you are connected to your instance.

Option 3: Connect from Mac or Linux Using an SSH Client

Your Mac or Linux computer most likely includes 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 ht-tp://www.openssh.org.

Open your command shell and run the following command:

```
$ ssh -i /path/key_pair.pem ec2-user@public_dns_name
```

> **Tip**
> For Amazon Linux, the user name is `ec2-user`. For RHEL5, the user name is either `root` or `ec2-user`. For Ubuntu, the user name is `ubuntu`. For Fedora, the user name is either `fedora` or `ec2-user`. For SUSE Linux, the user name is `root`. Otherwise, if `ec2-user` and `root` don't work, check with your AMI provider.

Add a Volume to Your Instance

Now that you've launched and connected to your Linux instance, you can run the following command on your instance to view its mounted volumes.

```
[ec2-user ~]$ 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 some room to install additional software on your instance (only 14% of the file system is being used above). For example, you can use the **yum** command to download and install packages.

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

To create and attach an Amazon EBS volume

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

2. In the navigation bar, select the region in which you created your instance, 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 **Availability Zone** for the instance.

3. In the navigation pane, under **Elastic Block Store**, click **Volumes**.

4. Click **Create Volume**.

5. Configure the following, and then click **Create**:

   - Select the **General Purpose (SSD)** volume type to create a General Purpose (SSD) EBS volume.

     Note
     Some AWS accounts created before 2012 might have access to Availability Zones in us-east-1, us-west-1, or ap-northeast-1 that do not support SSD volumes such as Provisioned IOPS (SSD) and General Purpose (SSD). If you are unable to create an SSD volume (or launch an instance with an SSD volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports General Purpose (SSD) and Provisioned IOPS (SSD) volumes by creating a 1 GiB General Purpose (SSD) volume in that zone.

   - Enter the size of the volume you want to create. The free tier benefits for Amazon EBS include up to 30 GiB of storage; therefore, to avoid being charged for this tutorial, choose a volume size that will keep you from exceeding that limit. For example, if the boot volume for the instance you created uses an 8 GiB Amazon EBS volume, then make sure to create a volume that is less than or equal to 22 GiB.

   - Select the same **Availability Zone** that you used when you created your instance. Otherwise, you can't attach the volume to your instance.

6. In the navigation pane, under **Elastic Block Store**, click **Volumes**. Notice that your newly created volume appears there and the state of the volume is `available`, so it's ready to be attached to an instance.

7. Right-click the newly created volume and select **Attach Volume**.

8. In the **Attach Volume** dialog box, configure the following, and then click **Attach**:

   - Start typing in the name or ID of your instance, then select it from the list of suggested options.

   - Specify an unused device name for that instance. We'll use `/dev/sdf` in this tutorial. If you select a different device name, be sure to note it as you'll need this information in the next procedure.

You'll notice that in the **Details** pane for your volume, the state of the volume is `in-use`, and the volume is attached to your instance with the device name `/dev/sdf`. However, if you return to your instance and run the **df -h** command again, you won't see the volume yet. That's because we need to mount the volume for **df -h** to see it. The **lsblk** command, however, can see all block devices attached to the instance.

> **Note**
> Some Linux distributions do not provide the **lsblk** command by default. If the **lsblk** command does not work, you can use **sudo fdisk -l | grep Disk** instead.

```
[ec2-user ~]$ lsblk
NAME     MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvdf     202:80   0   22G  0 disk
xvda1    202:1    0    8G  0 disk /
```

In the above example, **lsblk** reports that there are two block devices attached to the instance; `xvda1` is mounted as the root file system (note the `MOUNTPOINT` value of `/`) and `xvdf` is not mounted at all.

To make a volume available

1. Identify the device to mount. In the previous procedure, the new volume was attached to `/dev/sdf`. Depending on the block device drivers on your instance's operating system, the device may appear at a different location (such as `/dev/xvdf` in the previous example) than what you specified in the console (`/dev/sdf`); in some cases, even the trailing letter may change (for example, `/dev/xvdj`). Amazon Linux instances always create links from the device path that you specified in the console to the new device path, but other distributions (such as Ubuntu or Red Hat) are not as predictable.

   Use the **lsblk** command to list the available devices.

   > **Note**
   > Some Linux distributions do not provide the **lsblk** command by default. If the **lsblk** command does not work, you can use **sudo fdisk -l | grep Disk** instead.

   ```
   [ec2-user ~]$ lsblk
   NAME     MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
   xvdf     202:80   0   22G  0 disk
   xvda1    202:1    0    8G  0 disk /
   ```

   In the above example, the `xvdf` device is not mounted. Sometimes when you create a volume from a snapshot, the data on the volume is contained in a partition (such as `/dev/xvdf1`) instead of the root of the volume. In such a case, you would mount the `/dev/xvdf1` partition (the **lsblk** command output omits the `/dev/` portion of the file path). In this example, there is an empty volume with no partition, so you will mount `/dev/xvdf`.

2. Because you created an empty volume instead of restoring a volume from a snapshot in the previous procedure, you need to format the volume using **mkfs** before you can mount it. Use the following command to create an ext4 file system on the volume. Substitute the device name you used if you did not use `/dev/xvdf` when you attached the volume.

   > **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 volume that was restored from a snapshot), 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. For more information, see Making the Volume Available on Linux.

   > **Note**
   > SUSE Linux Enterprise Server 11 does not fully support ext4 file systems. If you chose a SLES 11 AMI for your instance, use `ext3` in the following command instead.

```
[ec2-user ~]$ sudo mkfs -t ext4 /dev/xvdf
```

3. To mount the device as `/mnt/my-data`, run the following commands.

```
[ec2-user ~]$ sudo mkdir /mnt/my-data
[ec2-user ~]$ sudo mount /dev/xvdf /mnt/my-data
```

    Be sure to specify the device name you identified in Step 1 (p. 33); otherwise, you might receive the following error when you run this mount command: "`mount: you must specify the filesystem type`". If you see this error, repeat Step 1 (p. 33) and use the correct device path (remember to add the `/dev/` to the device name you get from the **lsblk** command).

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

```
[ec2-user ~]$ df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/xvda1            7.9G  1.1G  6.8G  14% /
tmpfs                 298M     0  298M   0% /dev/shm
/dev/xvdf              22G     0   22G   0% /mnt/my-data
```

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

```
[ec2-user ~]$ ls /mnt/my-data
```

At this point, you have completed the example architecture for this tutorial. You can continue to customize and use your instance for as long as you wish.

Important
Remember, if you stayed within the free tier benefits, there are no charges. Otherwise, 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`.

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, Clean Up Your Instance and Volume (p. 34).

Clean Up Your Instance and Volume

After you've finished with the instance and the Amazon EBS volume that you created for this tutorial, you should clean up. First, terminate the instance, which detaches the volume from the instance, and then delete the volume.

Terminating an instance effectively deletes it because you can't reconnect to an 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. For more information about the differences between stopping and terminating an instance, see Stopping Instances.

To terminate the instance

1. Locate your instance in the list of instances on the **Instances** page. If you can't find your instance, verify that you have selected the correct region.

2. Right-click the instance, and then click **Terminate**.

3. Click **Yes, Terminate** when prompted for confirmation.

EBS volumes can persist even after your instance is terminated. 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 benefits of the free 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. Locate the volume that you created in the list of volumes on the **Volumes** page. If you can't find your volume, verify that you have selected the correct region.

2. Right-click the volume, and then click **Delete Volume**.

3. Click **Yes, Delete** when prompted for confirmation.
   Amazon EC2 begins deleting the volume.

Best Practices for Amazon EC2

This checklist is intended to help you get the maximum benefit from and satisfaction with Amazon EC2.

Security and Network

- Manage access to AWS resources and APIs using identity federation, IAM users, and IAM roles. Establish credential management policies and procedures for creating, distributing, rotating, and revoking AWS access credentials. For more information, see IAM Best Practices in the *Using IAM* guide.
- Implement the least permissive rules for your security group. For more information, see Security Group Rules (p. 423).
- Regularly patch, update, and secure the operating system and applications on your instance. For more information about updating Amazon Linux, see Managing Software on Your Linux Instance (p. 322).
- Launch your instances into a VPC instead of EC2-Classic. Note that if you created your AWS account after 2013-12-04, we automatically launch your instances into a VPC. For more information about the benefits, see Amazon EC2 and Amazon Virtual Private Cloud (VPC) (p. 468).

Storage

- Understand the implications of the root device type for data persistence, backup, and recovery. For more information, see Storage for the Root Device (p. 55).
- Use separate Amazon EBS volumes for the operating system versus your data. Ensure that the volume with your data persists after instance termination.
- Use the instance store available for your instance to store temporary data. Remember that the data stored in instance store is deleted when you stop or terminate your instance. If you use instance store for database storage, ensure that you have a cluster with a replication factor that ensures fault tolerance.

Resource Management

- Use instance metadata and custom resource tags to track and identify your AWS resources. For more information, see Instance Metadata and User Data (p. 256) and Tagging Your Amazon EC2 Resources (p. 610).
- View your current limits for Amazon EC2. Plan to request any limit increases in advance of the time that you'll need them. For more information, see Amazon EC2 Service Limits (p. 618).

Backup and Recovery

- Regularly back up your instance using Amazon EBS snapshots (p. 559) or a backup tool.

- Deploy critical components of your application across multiple Availability Zones, and replicate your data appropriately.
- Design your applications to handle dynamic IP addressing when your instance restarts. For more information, see Amazon EC2 Instance IP Addressing (p. 480).
- Monitor and respond to events. For more information, see Monitoring Amazon EC2 (p. 343).
- Ensure that you are prepared to handle failover. For a basic solution, you can manually attach a network interface or Elastic IP address to a replacement instance. For more information, see Elastic Network Interfaces (ENI) (p. 494). For an automated solution, you can use Auto Scaling. For more information, see the Auto Scaling Developer Guide.
- Regularly test the process of recovering your instances and Amazon EBS volumes if they fail.

Tutorial: Installing a LAMP Web Server

The following procedures help you install the Apache web server with PHP and MySQL support on your Amazon EC2 instance (sometimes called a LAMP web server or LAMP stack). You can use this server to host a static website or deploy a dynamic PHP application that reads and writes information to a database.

Prerequisites

This tutorial assumes that you have already launched an instance with a public DNS name that is reachable from the Internet. For more information, see Launch an Amazon EC2 Instance (p. 27). You must also have configured your security group to allow `SSH` (port 22), `HTTP` (port 80), and `HTTPS` (port 443) connections. For more information about these prerequisites, see Setting Up with Amazon EC2 (p. 20).

> **Important**
> These procedures are intended for use with Amazon Linux, but the commands and file locations are similar for Red Hat and CentOS. For more information about other distributions, see their specific documentation. **If you are trying to set up a LAMP web server on an Ubuntu instance, this tutorial will not work for you.** For information about LAMP web servers on Ubuntu, go to the Ubuntu community documentation ApacheMySQLPHP topic.

To install and start the LAMP web server

1. Connect to your instance (p. 28).
2. To ensure that all of your software packages are up to date, perform a quick software update on your instance. This process may take a few minutes, but it is important to make sure you have the latest security updates and bug fixes.

   > **Note**
   > The `-y` option installs the updates without asking for confirmation. If you would like to examine the updates before installing, you can omit this option.

   ```
   [ec2-user ~]$ sudo yum update -y
   ```

3. Now that your instance is current, you can install the Apache web server, MySQL, and PHP software packages. Use the **yum groupinstall** command to install multiple software packages and all related dependencies at the same time.

```
[ec2-user ~]$ sudo yum groupinstall -y "Web Server" "MySQL Database" "PHP
Support"
```

> **Note**
> Non-Amazon Linux instances may have subtle differences in their group names. If the above command fails because of an invalid group name, use the **yum grouplist** command and scan the output for similar groups, such as "MySQL Database server" instead of "MySQL Database", and use the appropriate group name for your distribution.

4. Install the `php-mysql` package.

```
[ec2-user ~]$ sudo yum install -y php-mysql
```

5. Start the Apache web server.

```
[ec2-user ~]$ sudo service httpd start
Starting httpd:                                            [  OK  ]
```

6. Use the **chkconfig** command to configure the Apache web server to start at each system boot.

```
[ec2-user ~]$ sudo chkconfig httpd on
```

> **Tip**
> The **chkconfig** command does not provide any confirmation message when you successfully enable a service. You can verify that **httpd** is on by running the following command.

```
[ec2-user ~]$ chkconfig --list httpd
httpd           0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

Here, **httpd** is on in runlevels 2, 3, 4, and 5 (which is what you want to see).

7. Test your web server. In a web browser, enter the public DNS address (or the public IP address) of your instance; you should see the Apache test page. You can get the public DNS for your instance using the Amazon EC2 console (check the **Public DNS** column; if this column is hidden, click the **Show/Hide** icon and select **Public DNS**).

> **Tip**
> If you are unable to see the Apache test page, check that the security group you are using contains a rule to allow HTTP (port 80) traffic. For information about adding an HTTP rule to your security group, see Adding Rules to a Security Group (p. 426).

> **Important**
> If you are not using Amazon Linux, you may also need to configure the firewall on your instance to allow these connections. For more information about how to configure the firewall, see the documentation for your specific distribution.

Amazon Linux AMI **Test Page**

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:

Powered by **APACHE** **2.2**

Note
This test page appears only when there is no content in `/var/www/html`. When you add content to the document root, your content appears at the public DNS address of your instance instead of this test page.

Apache **httpd** serves files that are kept in a directory called the Apache document root. The Amazon Linux Apache document root is `/var/www/html`, which is owned by `root` by default.

```
[ec2-user ~]$ ls -l /var/www
total 16
drwxr-xr-x 2 root root 4096 Jul 12 01:00 cgi-bin
drwxr-xr-x 3 root root 4096 Aug  7 00:02 error
drwxr-xr-x 2 root root 4096 Jan  6  2012 html
drwxr-xr-x 3 root root 4096 Aug  7 00:02 icons
```

To allow `ec2-user` to manipulate files in this directory, you need to modify the ownership and permissions of the directory. There are many ways to accomplish this task; in this tutorial, you add a `www` group to your instance, and you give that group ownership of the `/var/www` directory and add write permissions for the group. Any members of that group will then be able to add, delete, and modify files for the web server.

To set file permissions

1. Add the `www` group to your instance.

    ```
    [ec2-user ~]$ sudo groupadd www
    ```

2. Add your user (in this case, `ec2-user`) to the `www` group.

    ```
    [ec2-user ~]$ sudo usermod -a -G www ec2-user
    ```

> **Important**
> You need to log out and log back in to pick up the new group. You can use the **exit** command, or close the terminal window.

3. Log out and then log back in again, and verify your membership in the www group.

   a. Log out.

   ```
   [ec2-user ~]$ exit
   ```

   b. Reconnect to your instance, and then run the following command to verify your membership in the www group.

   ```
   [ec2-user ~]$ groups
   ec2-user wheel www
   ```

4. Change the group ownership of /var/www and its contents to the www group.

   ```
   [ec2-user ~]$ sudo chown -R root:www /var/www
   ```

5. Change the directory permissions of /var/www and its subdirectories to add group write permissions and to set the group ID on future subdirectories.

   ```
   [ec2-user ~]$ sudo chmod 2775 /var/www
   [ec2-user ~]$ find /var/www -type d -exec sudo chmod 2775 {} +
   ```

6. Recursively change the file permissions of /var/www and its subdirectories to add group write permissions.

   ```
   [ec2-user ~]$ find /var/www -type f -exec sudo chmod 0664 {} +
   ```

Now ec2_user (and any future members of the www group) can add, delete, and edit files in the Apache document root. Now you are ready to add content, such as a static website or a PHP application.

To test your LAMP web server

If your server is installed and running, and your file permissions are set correctly, your ec2-user account should be able to create a simple PHP file in the /var/www/html directory that will be available from the Internet.

1. Create a simple PHP file in the Apache document root.

   ```
   [ec2-user ~]$ echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
   ```

   > **Tip**
   > If you get a "Permission denied" error when trying to run this command, try logging out and logging back in again to pick up the proper group permissions that you configured in To set file permissions (p. 40).

2. In a web browser, enter the URL of the file you just created. This URL is the public DNS address of your instance followed by a forward slash and the file name. For example:

    ```
    http://my.public.dns.amazonaws.com/phpinfo.php
    ```

    You should see the PHP information page.

    Note
    If you do not see the above image, but instead see the `phpinfo()` source code, verify that the `php` package was installed properly. Your output may look slightly different, but look for `php` in the `Installed Packages` section.

    ```
    [ec2-user wordpress]$ yum list installed php
    Loaded plugins: priorities, security, update-motd, upgrade-helper
    amzn-main                                              | 2.1 kB
      00:00
    amzn-updates                                           | 2.3 kB
      00:00
    Installed Packages
    php.x86_64                          5.3.27-1.0.amzn1
    @amzn-updates
    ```

    If this package is not listed as installed, install it with the following command, restart the **httpd** service, and reload the `phpinfo.php` page.

    ```
    [ec2-user wordpress]$ sudo yum install -y php
    [ec2-user wordpress]$ sudo service httpd restart
    ```

3. Delete the `phpinfo.php` file. Although this can be useful information to you, it should not be broadcast to the Internet for security reasons.

    ```
    [ec2-user ~]$ rm /var/www/html/phpinfo.php
    ```

To secure the MySQL server

The default installation of the MySQL server has several features that are great for testing and development, but they should be disabled or removed for production servers. The **mysql_secure_installation** command walks you through the process of setting a root password and removing the insecure features from your installation. Even if you are not planning on using the MySQL server, performing this procedure is a good idea.

1. Start the MySQL server so that you can run **mysql_secure_installation**.

    ```
    [ec2-user ~]$ sudo service mysqld start
    Initializing MySQL database:  Installing MySQL system tables...
    OK
    Filling help tables...
    OK

    To start mysqld at boot time you have to copy
    support-files/mysql.server to the right place for your system

    PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
    ...

    Starting mysqld:                                           [  OK  ]
    ```

2. Run **mysql_secure_installation**.

    ```
    [ec2-user ~]$ sudo mysql_secure_installation
    ```

    a. When prompted, enter a password for the `root` account.

        i.   Enter the current `root` password. By default, the `root` account does not have a password set, so press **Enter**.
        ii. Type **Y** to set a password, and enter a secure password twice. For more information about creating a secure password, go to http://www.pctools.com/guides/password/. Make sure to store this password in a safe place.

    b. Type **Y** to remove the anonymous user accounts.
    c. Type **Y** to disable remote `root` login.
    d. Type **Y** to remove the test database.
    e. Type **Y** to reload the privilege tables and save your changes.

3. (Optional) Stop the MySQL server if you do not plan to use it right away. You can restart the server when you need it again.

    ```
    [ec2-user ~]$ sudo service mysqld stop
    Stopping mysqld:                                           [  OK  ]
    ```

4. (Optional) If you want the MySQL server to start at every boot, enter the following command.

    ```
    [ec2-user ~]$ sudo chkconfig mysqld on
    ```

You should now have a fully functional LAMP web server. If you add content to the Apache document root at `/var/www/html`, you should be able to view that content at the public DNS address for your instance.

Related Topics

For more information on transferring files to your instance or installing a WordPress blog on your web server, see the following topics:

- Transferring Files to Your Linux Instance Using WinSCP (p. 306)
- Transferring Files to Linux Instances from Linux Using SCP (p. 301)
- Tutorial: Hosting a WordPress Blog with Amazon EC2 (p. 45)

For more information about the Apache web server, go to http://httpd.apache.org/. For more information about the MySQL database server, go to http://www.mysql.com/. For more information about the PHP programming language, go to http://php.net/.

If you are interested in registering a domain name for your web server, or transferring an existing domain name to this host, see Creating and Migrating Domains and Subdomains to Amazon Route 53 in the *Amazon Route 53 Developer Guide*.

Tutorial: Hosting a WordPress Blog with Amazon EC2

The following procedures will help you install, configure, and secure a WordPress blog on your Amazon EC2 instance.

> **Important**
> These procedures are intended for use with Amazon Linux, but the commands and file locations are similar for Red Hat and CentOS. For more information about other distributions, see their specific documentation.

This tutorial is a good introduction to using Amazon EC2 in that you have full control over a web server that hosts your WordPress blog, which is not typical with a traditional hosting service. Of course, that means that you are responsible for updating the software packages and maintaining security patches for your server as well. For a more automated WordPress installation that does not require direct interaction with the web server configuration, the AWS CloudFormation service provides a WordPress template that can also get you started quickly. For more information, see Get Started in the *AWS CloudFormation User Guide*. If you'd prefer to host your WordPress blog on a Windows instance, see Deploying a WordPress Blog on Your Amazon EC2 Windows Instance in the *Amazon EC2 User Guide for Microsoft Windows Instances*.

Prerequisites

This tutorial assumes that you have launched an instance with a functional web server with PHP and MySQL support. Your Amazon EC2 security group should also allow `HTTP` and `HTTPS` traffic. If you do not already have a functional web server, see Tutorial: Installing a LAMP Web Server (p. 38) to create one and then return to this tutorial to install WordPress. For information about adding rules to your security group, see Adding Rules to a Security Group (p. 426).

To download and unzip the WordPress installation package

1. Download the latest WordPress installation package with the **wget** command. The following command should always download the latest release.

    ```
    [ec2-user ~]$ wget https://wordpress.org/latest.tar.gz
    --2013-08-09 17:19:01--  https://wordpress.org/latest.tar.gz
    Resolving wordpress.org (wordpress.org)... 66.155.40.249, 66.155.40.250
    Connecting to wordpress.org (wordpress.org)|66.155.40.249|:443... connected.
    HTTP request sent, awaiting response... 200 OK
    ```

```
Length: 4028740 (3.8M) [application/x-gzip]
Saving to: latest.tar.gz

100%[======================================>] 4,028,740   20.1MB/s   in 0.2s

2013-08-09 17:19:02 (20.1 MB/s) - latest.tar.gz saved [4028740/4028740]
```

2. Unzip and unarchive the installation package. The installation folder is unzipped to a folder called
    `wordpress`.

```
[ec2-user ~]$ tar -xzf latest.tar.gz
[ec2-user ~]$ ls
latest.tar.gz   wordpress
```

To create a MySQL user and database for your WordPress installation

Your WordPress installation needs to store information, such as blog post entries and user comments,
in a database. This procedure will help you create a database for your blog and a user that is authorized
to read and save information to that database.

1. Start the MySQL server.

```
[ec2-user ~]$ sudo service mysqld start
```

2. Log in to the MySQL server as the `root` user. Enter your MySQL `root` password when prompted;
    this may be different than your `root` system password, or it may even be empty if you have not se-
    cured your MySQL server.

    Important
    If you have not secured your MySQL server yet, it is very important that you do so. For more
    information, see To secure the MySQL server (p. 43).

```
[ec2-user ~]$ mysql -u root -p
Enter password:
```

3. Create a user and password for your MySQL database. Your WordPress installation uses these
    values to communicate with your MySQL database. Enter the following command, substituting a
    unique user name and password.

```
mysql> CREATE USER 'wordpress-user'@'localhost' IDENTIFIED BY
'your_strong_password';
Query OK, 0 rows affected (0.00 sec)
```

    Make sure that you create a strong password for your user. Do not use the single quote character (
    ') in your password, because this will break the preceding command. For more information about
    creating a secure password, go to http://www.pctools.com/guides/password/. Do not reuse an existing
    password, and make sure to store this password in a safe place.

4. Create your database. Give your database a descriptive, meaningful name, such as `wordpress-
    db`.

Note

The punctuation marks surrounding the database name in the command below are called backticks. The backtick (`` ` ``) key is usually located above the **Tab** key on a standard keyboard. Backticks are not always required, but they allow you to use otherwise illegal characters, such as hyphens, in database names.

```
mysql> CREATE DATABASE `wordpress-db`;
Query OK, 1 row affected (0.01 sec)
```

5. Grant full privileges for your database to the WordPress user you created earlier.

```
mysql> GRANT ALL PRIVILEGES ON `wordpress-db`.* TO "wordpress-user"@"local
host";
Query OK, 0 rows affected (0.00 sec)
```

6. Flush the MySQL privileges to pick up all of your changes.

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)
```

7. Exit the `mysql` client.

```
mysql> exit
Bye
```

To create and edit the wp-config.php file

The WordPress installation folder contains a sample configuration file called `wp-config-sample.php`. In this procedure, you copy this file and edit it to fit your specific configuration.

1. Copy the `wp-config-sample.php` file to a file called `wp-config.php`. This creates a new configuration file and keeps the original sample file intact as a backup.

```
[ec2-user ~]$ cd wordpress/
[ec2-user wordpress]$ cp wp-config-sample.php wp-config.php
```

2. Edit the `wp-config.php` file with your favorite text editor (such as **nano** or **vim**) and enter values for your installation. If you do not have a favorite text editor, `nano` is much easier for beginners to use.

```
[ec2-user wordpress]$ nano wp-config.php
```

a. Find the line that defines `DB_NAME` and change `database_name_here` to the database name you created in of .

```
define('DB_NAME', 'wordpress-db');
```

b. Find the line that defines `DB_USER` and change `username_here` to the database user you created in Step 3 (p. 46) of To create a MySQL user and database for your WordPress installation (p. 46).

```
define('DB_USER', 'wordpress-user');
```

c. Find the line that defines `DB_PASSWORD` and change `password_here` to the strong password you created in Step 3 (p. 46) of To create a MySQL user and database for your WordPress installation (p. 46).

```
define('DB_PASSWORD', 'your_strong_password');
```

d. Find the section called `Authentication Unique Keys and Salts`. These `KEY` and `SALT` values provide a layer of encryption to the browser cookies that WordPress users store on their local machines. Basically, adding long, random values here makes your site more secure. Visit https://api.wordpress.org/secret-key/1.1/salt/ to randomly generate a set of key values that you can copy and paste into your `wp-config.php` file. To paste text into a PuTTY terminal, place the cursor where you want to paste the text and right-click your mouse inside the PuTTY terminal.

   For more information about security keys, go to http://codex.wordpress.org/Editing_wp-config.php#Security_Keys.

   > **Note**
   > The values below are for example purposes only; do not use these values for your installation.

```
define('AUTH_KEY',          ' #U$$+[RXN8:b^-L 0(WU_+ c+WFkI~c]o]-
bHw+)/Aj[wTwSiZ<Qb[mghEXcRh-');
define('SECURE_AUTH_KEY',  'Zsz._P=l/|y.Lq)XjlkwS1y5NJ76E6EJ.AV0pCK
ZZB,*~*r ?6OP$eJT@;+(ndLg');
define('LOGGED_IN_KEY',    'ju}qwre3V*+8f_zOWf?{LlGsQ]Ye@2Jh^,8x>)Y
|;(^[Iw]Pi+LG#A4R?7N`YB3');
define('NONCE_KEY',
'P(g62HeZxEes|LnI^i=H,[XwK9I&[2s|:?0N}VJM%?;v2v]v+;+^9eXUahg@::Cj');
define('AUTH_SALT',
'C$DpB4Hj[JK:?{ql`sRVa:{:7yShy(9A@5wg+`JJVb1fk%_-Bx*M4(qc[Qg%JT!h');
define('SECURE_AUTH_SALT',
'd!uRu#}+q#{f$Z?Z9uFPG.${+S{n~1M&%@~gL>U>NV<zpD-@2-Es7Q1O-bp28EKv');
define('LOGGED_IN_SALT',   ';j{00P*owZf)kVD+FVLn-~
>.|Y%Ug4#I^*LVd9QeZ^&XmK|e(76miC+&W&+^0P/');
define('NONCE_SALT',
'-97r*V/cgxLmp?Zy4zUU4r99QQ_rGs2LTd%P;|_e1tS)8_B/,.6[=UK<J_y9?JWG');
```

e. Save the file and exit your text editor.

To move your WordPress installation to the Apache document root

Now that you've unzipped the installation folder, created a MySQL database and user, and customized the WordPress configuration file, you are ready to move your installation files to your web server document root so you can run the installation script that completes your installation. The location of these files depends on whether you want your WordPress blog to be available at the root of your web server (for example, *my.public.dns.amazonaws.com*) or in a subdirectory or folder (for example, *my.public.dns.amazonaws.com/blog*).

Important

Choose the location where you want your blog to be available and only run the **mv** associated with that location. If you run both sets of commands below, you will get an error message on the second **mv** command because the files you are trying to move are no longer there.

1. To make your blog available at *my.public.dns.amazonaws.com*, move the files in the `wordpress` folder (but not the folder itself) to the Apache document root (`/var/www/html` on Amazon Linux instances).

```
[ec2-user wordpress]$ mv * /var/www/html/
```

2. *OR*, to make your blog available at *my.public.dns.amazonaws.com/blog* instead, create a new folder called `blog` inside the Apache document root and move the files in the `wordpress` folder (but not the folder itself) to the new `blog` folder.

```
[ec2-user wordpress]$ mkdir /var/www/html/blog
[ec2-user wordpress]$ mv * /var/www/html/blog
```

Important

If you are not moving on to the next procedure immediately, stop the Apache web server (`httpd`) now for security purposes. After you move your installation to the Apache document root, the WordPress installation script is unprotected and an attacker could gain access to your blog if the Apache web server were running. To stop the Apache web server, enter the command **sudo service httpd stop**. If you are moving on to the next procedure, you do not need to stop the Apache web server.

To fix file permissions for the Apache web server

Some of the available features in WordPress require write access to the Apache document root (such as uploading media though the Administration screens). The web server runs as the `apache` user, so you need to add that user to the `www` group that was created in the LAMP web server tutorial (p. 38).

1. Add the `apache` user to the `www` group.

```
[ec2-user wordpress]$ sudo usermod -a -G www apache
```

2. Change the group ownership of `/var/www` and its contents to the `www` group.

```
[ec2-user wordpress]$ sudo chgrp -R www /var/www
```

3. Change the directory permissions of `/var/www` and its subdirectories to add group write permissions and to set the group ID on future subdirectories.

```
[ec2-user wordpress]$ sudo chmod 2775 /var/www
[ec2-user wordpress]$ find /var/www -type d -exec sudo chmod 2775 {} +
```

4. Recursively change the file permissions of `/var/www` and its subdirectories to add group write permissions.

```
[ec2-user wordpress]$ find /var/www -type f -exec sudo chmod 0664 {} +
```

5.   Restart the Apache web server to pick up the new group and permissions.

```
[ec2-user wordpress]$ sudo service httpd restart
Stopping httpd:                                          [  OK  ]
Starting httpd:                                          [  OK  ]
```

To run the WordPress installation script

1.   Use the **chkconfig** command to ensure that the `httpd` and `mysqld` services start at every system boot.

```
[ec2-user wordpress]$ sudo chkconfig httpd on
[ec2-user wordpress]$ sudo chkconfig mysqld on
```

2.   Verify that the MySQL server (`mysqld`) is running.

```
[ec2-user wordpress]$ sudo service mysqld status
mysqld (pid  4746) is running...
```

If the `mysqld` service is not running, start it.

```
[ec2-user wordpress]$ sudo service mysqld start
Starting mysqld:                                         [  OK  ]
```

3.   Verify that your Apache web server (`httpd`) is running.

```
[ec2-user wordpress]$ sudo service httpd status
httpd (pid  502) is running...
```

If the `httpd` service is not running, start it.

```
[ec2-user wordpress]$ sudo service httpd start
Starting httpd:                                          [  OK  ]
```

4.   Verify that the `php` and `php-mysql` packages are installed. Your output may look slightly different, but look for the `Installed Packages` section.

```
[ec2-user wordpress]$ yum list installed php php-mysql
Loaded plugins: priorities, security, update-motd, upgrade-helper
amzn-main                                               | 2.1 kB     00:00
amzn-updates                                            | 2.3 kB     00:00
Installed Packages
php.x86_64                    5.3.27-1.0.amzn1                    @amzn-
updates
php-mysql.x86_64              5.3.27-1.0.amzn1                    @amzn-
updates
```

> **Note**
> If either of these packages are not listed as installed, install them with the following command and then restart the **httpd** service.

```
[ec2-user wordpress]$ sudo yum install -y php php-mysql
[ec2-user wordpress]$ sudo service httpd restart
```

5. In a web browser, enter the URL of your WordPress blog (either the public DNS address for your instance, or that address followed by the `blog` folder). You should see the WordPress installation screen.

```
http://my.public.dns.amazonaws.com
```
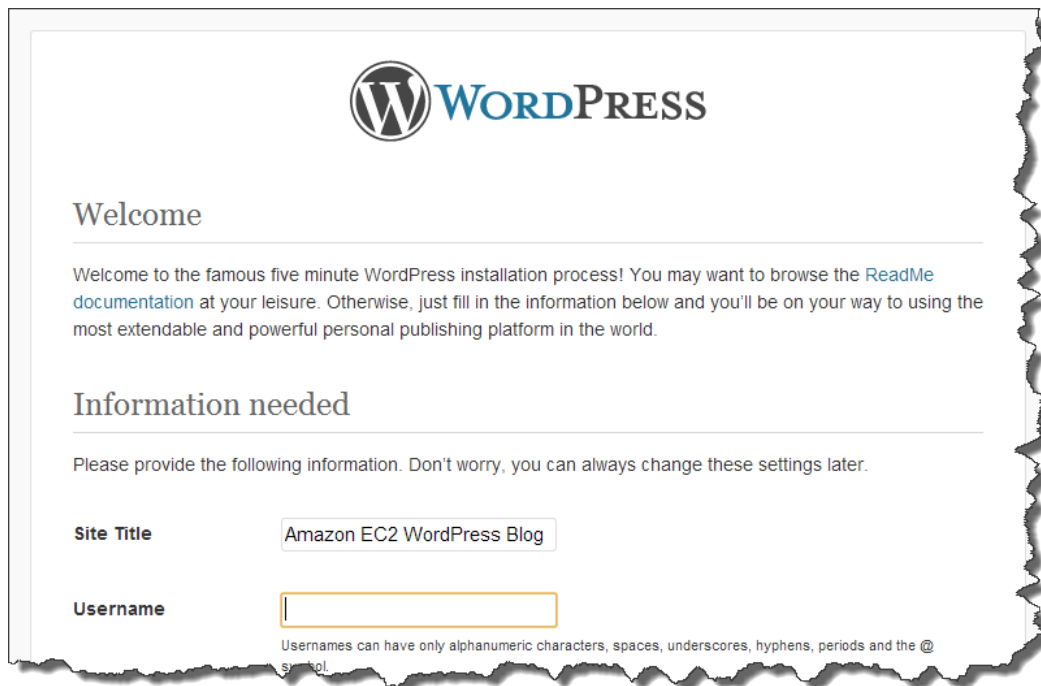

6. Enter the remaining installation information into the WordPress installation wizard.

Field	Value
Site Title	Enter a name for your WordPress site.
Username	Enter a name for your WordPress administrator. For security purposes you should choose a unique name for this user, since this will be more difficult to exploit than the default user name, `admin`.
Password	Enter a strong password, and then enter it again to confirm. Do not reuse an existing password, and make sure to store this password in a safe place.
Your E-mail	Enter the email address you want to use for notifications.

7. Click **Install WordPress** to complete the installation.

Congratulations, you should now be able to log into your WordPress blog and start posting entries.

If your WordPress blog becomes popular and you need more compute power, you might consider migrating to a larger instance type; for more information, see Resizing Your Instance (p. 128). If your blog requires more storage space than you originally accounted for, you could expand the storage space on your instance; for more information, see Expanding the Storage Space of a Volume (p. 545). If your MySQL database needs to grow, you could consider moving it to Amazon RDS to take advantage of the service's autoscaling abilities.

For information about WordPress, see the WordPress Codex help documentation at http://codex.word-press.org/. For more information about troubleshooting your installation, go to http://codex.wordpress.org/Installing_WordPress#Common_Installation_Problems. For information about making your WordPress blog more secure, go to http://codex.wordpress.org/Hardening_WordPress. For information about keeping your WordPress blog up-to-date, go to http://codex.wordpress.org/Updating_WordPress.

Amazon Machine Images (AMI)

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

An AMI includes the following:

- A template for the root volume for the instance (for example, an operating system, an application server, and applications)
- Launch permissions that control which AWS accounts can use the AMI to launch instances
- A block device mapping that specifies the volumes to attach to the instance when it's launched

Using an AMI

The following diagram summarizes the AMI lifecycle. After you create and register an AMI, you can use it to launch new instances. (You can also launch instances from an AMI if the AMI owner grants you launch permissions.) You can copy an AMI to the same region or to different regions. When you are finished launching instance from an AMI, you can deregister the AMI.
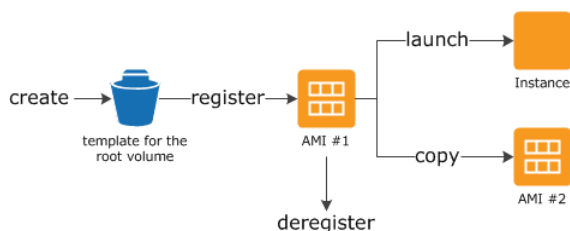
You can search for an AMI that meets the criteria for your instance. You can search for AMIs provided by AWS or AMIs provided by the community. For more information, see AMI Types (p. 55) and Finding a Linux AMI (p. 58).

When you are connected to an instance, you can use it just like you use any other server. For information about launching, connecting, and using your instance, see Amazon EC2 Instances (p. 103).

Creating Your Own AMI

You can customize the instance that you launch from a public AMI and then save that configuration as a custom AMI for your own use. Instances that you launch from your AMI use all the customizations that you've made.

The root storage device of the instance determines the process you follow to create an AMI. The root volume of an instance is either an Amazon EBS volume or an instance store volume. For information, see Amazon EC2 Root Device Volume (p. 14).

To create an Amazon EBS-backed AMI, see Creating an Amazon EBS-Backed Linux AMI (p. 74). To create an instance store-backed AMI, see Creating an Instance Store-Backed Linux AMI (p. 77).

To help categorize and manage your AMIs, you can assign custom *tags* to them. For more information, see Tagging Your Amazon EC2 Resources (p. 610).

Buying, Sharing, and Selling AMIs

After you create an AMI, you can keep it private so that only you can use it, or you can share it with a specified list of AWS accounts. You can also make your custom AMI public so that the community can use it. Building a safe, secure, usable AMI for public consumption is a fairly straightforward process, if you follow a few simple guidelines. For information about how to create and use shared AMIs, see Shared AMIs (p. 60).

You can purchase an AMIs from a third party, including AMIs that come with service contracts from organizations such as Red Hat. You can also create an AMI and sell it to other Amazon EC2 users. For more information about buying or selling AMIs, see Paid AMIs (p. 70).

Deregistering Your AMI

You can deregister an AMI when you have finished with it. After you deregister an AMI, you can't use it to launch new instances. For more information, see Deregistering Your AMI (p. 90).

Amazon Linux

The Amazon Linux AMI is a supported and maintained Linux image provided by AWS. The following are some of the features of Amazon Linux:

* A stable, secure, and high-performance execution environment for applications running on Amazon EC2.
* Provided at no additional charge to Amazon EC2 users.
* An Amazon EBS-backed, PV-GRUB AMI that includes Linux 3.4, AWS tools, and repository access to multiple versions of MySQL, PostgreSQL, Python, Ruby, and Tomcat.
* Updated on a regular basis to include the latest components, and these updates are also made available in the **yum** repositories for installation on running instances.
* Includes packages that enable easy integration with AWS services, such as the Amazon EC2 API and AMI tools, the Boto library for Python, and the Elastic Load Balancing tools.

For more information, see Amazon Linux (p. 91).

AMI Types

You can select an AMI to use based on the following characteristics:

- Region (see Regions and Availability Zones (p. 7))
- Operating system
- Architecture (32-bit or 64-bit)
- Launch Permissions (p. 55)
- Storage for the Root Device (p. 55)

Launch Permissions

The owner of an AMI determines its availability by specifying launch permissions. Launch permissions fall into the following categories.

Launch Permission	Description
public	The owner grants launch permissions to all AWS accounts.
explicit	The owner grants launch permissions to specific AWS accounts.
implicit	The owner has implicit launch permissions for an AMI.

Amazon and the Amazon EC2 community provide a large selection of public AMIs. For more information, see Shared AMIs (p. 60). Developers can charge for their AMIs. For more information, see Paid AMIs (p. 70).

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. 14).

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.

Characteristic	Amazon EBS-Backed	Amazon Instance Store-Backed
Upgrading	The instance type, kernel, RAM disk, and user data can be changed while the instance is stopped.	Instance attributes are fixed for the life of an instance.
Charges	You're charged for instance usage, Amazon EBS volume usage, and storing your AMI as an Amazon EBS snapshot.	You're charged for instance usage and storing your AMI in Amazon S3.
AMI creation/bundling	Uses a single command/call	Requires installation and use of AMI tools
Stopped state	Can be placed in stopped state where instance is not running, but the root volume is persisted in Amazon EBS	Cannot be in stopped state; instances are running or terminated

* By default, Amazon EBS-backed instance root volumes have the `DeleteOnTermination` flag set to `true`, which causes the volume to be deleted upon instance termination. For information about how to change this so that the volume persists following termination, see Changing the Root Device Volume to Persist (p. 17).

Determining the Root Device Type of Your AMI

To determine the root device type of an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**, and select the AMI.
3. Check the value of **Root Device Type** in the **Details** tab as follows:

   - If the value is `ebs`, this is an Amazon EBS-backed AMI.
   - If the value is `instance store`, this is an instance store-backed AMI.

To determine the root device type of an AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-images (AWS CLI)
- ec2-describe-images (Amazon EC2 CLI)
- Get-EC2Image (AWS Tools for Windows PowerShell)

Stopped State

You can stop an Amazon EBS-backed instance, but not an Amazon EC2 instance store-backed instance. Stopping causes the instance to stop running (its status goes from `running` to `stopping` to `stopped`). A stopped instance persists in Amazon EBS, which allows it to be restarted. Stopping is different from terminating; you can't restart a terminated instance. Because Amazon EC2 instance store-backed AMIs can't be stopped, they're either running or terminated. For more information about what happens and what you can do while an instance is stopped, see Stop and Start Your Instance (p. 309).

Default Data Storage and Persistence

Instances that use an instance store volume for the root device automatically have instance store available (the root volume contains the root partition and you can store additional data). Any data on an instance store volume is deleted when the instance fails or terminates (except for data on the root device). You can add persistent storage to your instance by attaching one or more Amazon EBS volumes.

Instances that use Amazon EBS for the root device automatically have an Amazon EBS volume attached. The volume appears in your list of volumes like any other. The instances don't use any available instance store volumes by default. You can add instance storage or additional Amazon EBS volumes using a block device mapping. For more information, see Block Device Mapping (p. 591). For information about what happens to the instance store volumes when you stop an instance, see Stop and Start Your Instance (p. 309).

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 AMIs backed by instance store, you must create an AMI from your instance on the instance itself, but there aren't any API actions to help you.

AMI creation is much easier for AMIs backed by Amazon EBS. The `CreateImage` API action creates your Amazon EBS-backed AMI and registers it. There's also a button in the AWS Management Console that lets you create an AMI from a running instance. For more information, see Creating an Amazon EBS-Backed Linux AMI (p. 74).

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).

Virtualization Types

Amazon Machine Images use one of two types of virtualization: paravirtual (PV) or hardware virtual machine (HVM). All current generation instance types support HVM AMIs. Some previous generation instance types, such as T1, C1, M1, and M2 do not support Linux HVM AMIs. Some current generation instance types, such as T2, I2, R3, G2, CR1, and CC2 do not support PV AMIs. The main difference between PV and HVM AMIs is the way in which they boot and whether they can take advantage of special hardware extensions (CPU, network, and storage) for better performance.

> **Note**
> For the best performance, we recommend that you use current generation instance types and HVM AMIs when you launch new instances. For more information on current generation instance types, see the Amazon EC2 Instances detail page. If you are using previous generation instance types and you are curious about upgrade paths, see Upgrade Paths.

Paravirtual (PV)

Paravirtual AMIs boot with a special boot loader called PV-GRUB, which starts the boot cycle and then chain loads the kernel specified in the `menu.lst` file on your image. Paravirtual guests can run on host hardware that does not have explicit support for virtualization, but they cannot take advantage of special hardware extensions such as enhanced networking or GPU processing. Historically, PV guests had better performance than HVM guests in many cases, but because of enhancements in HVM virtualization and the availability of PV drivers for HVM AMIs, this is no longer true.

For more information on PV-GRUB and its use in Amazon EC2, see PV-GRUB (p. 98).

Hardware Virtual Machine (HVM)

HVM AMIs are presented with a fully virtualized set of hardware and boot by executing the master boot record of the root block device of your image. This virtualization type provides the ability to run an operating system directly on top of a virtual machine without any modification, as if it were run on the baremetal hardware. The Amazon EC2 host system emulates some or all of the underlying hardware that is presented to the guest.

Unlike PV guests, HVM guests can take advantage of hardware extensions that provide fast access to the underlying hardware on the host system. For more information on CPU virtualization extensions available in Amazon EC2, see Server Virtualization on the Intel website. HVM AMIs are required to take advantage of enhanced networking and GPU processing. In order to pass through instructions to specialized network and GPU devices, the OS needs to be able to have access to the native hardware platform; HVM virtualization provides this access. For more information, see Enhanced Networking (p. 507) and Linux GPU Instances (p. 115).

PV on HVM

Paravirtual guests traditionally performed better with storage and network operations than HVM guests because they could leverage special drivers for I/O that avoided the overhead of emulating network and disk hardware, whereas HVM guests had to translate these instructions to emulated hardware. Now these PV drivers are available for HVM guests, so operating systems that cannot be ported to run in a paravirtualized environment can still see performance advantages in storage and network I/O by using them. With these PV on HVM drivers, HVM guests can get the same, or better, performance than paravirtual guests.

Finding a Linux AMI

Before you can launch an instance, you must select an AMI to use. As you select an AMI, consider the following requirements you might have for the instances that you'll launch:

- The region
- The operating system
- The architecture: 32-bit (`i386`) or 64-bit (`x86_64`)
- The root device type: Amazon EBS or instance store
- The provider: Amazon Web Services, Oracle, IBM, Microsoft, or the community

If you need to find a Windows AMI, see Finding a Windows AMI in the *Amazon EC2 User Guide for Microsoft Windows Instances.*

Finding a Linux AMI Using the Amazon EC2 Console

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

To find a Linux AMI using the console

1. Open the Amazon EC2 console.
2. From the navigation bar, select a region. You can select any region that's available to you, regardless of your location. This is the region in which you'll launch your instance.
3. In the navigation pane, click **AMIs**.
4. (Optional) Use the **Filter** options to scope the list of displayed AMIs to the AMIs that interest you. For example, to list all Linux AMIs provided by AWS, select **Public images**, **Amazon images**, and then your Linux distribution from the **Filter** lists.
5. (Optional) Click the **Show/Hide Columns** icon to select which image attributes to display, such as the root device type. Alternatively, you can select an AMI from the list and view its properties in the **Details** tab.
6. Before you select an AMI, it's important that you check whether it's backed by instance store or by Amazon EBS and that you are aware of the effects of this difference. For more information, see Storage for the Root Device (p. 55).
7. To launch an instance from this AMI, select it and then click **Launch**. For more information about launching an instance using the console, see Launching Your Instance from an AMI (p. 292). If you're not ready to launch the instance now, write down the AMI ID (ami-*xxxxxxxx*) for later.

Finding an AMI Using the Command Line

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

- describe-images (AWS CLI)

```
$ aws ec2 describe-images --owners self amazon
```

- ec2-describe-images (Amazon EC2 CLI)

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

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

- Launching an Instance Using the AWS CLI in the *AWS Command Line Interface User Guide*
- Launching an Instance Using the Amazon EC2 CLI in the *Amazon EC2 Command Line Reference*
- Launching an Instance Using Windows PowerShell in the *AWS Tools for Windows PowerShell User Guide*

Shared AMIs

A shared AMI is an AMI that a developer created and made available for other developers to use. One of the easiest ways to get started with Amazon EC2 is to use a shared AMI that has the components you need and then add custom content.

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

We recommend that you get an AMI from a trusted source. If you have questions or observations about a shared AMI, use the AWS forums.

Amazon's public images have an aliased owner, which appears as `amazon` in the account field. This enables you to find AMIs from Amazon easily. Other users can't alias their AMIs.

Topics
- Finding Shared AMIs (p. 60)
- Making an AMI Public (p. 63)
- Sharing an AMI with Specific AWS Accounts (p. 64)
- Using Bookmarks (p. 66)
- Guidelines for Shared Linux AMIs (p. 66)

Finding Shared AMIs

You can use the Amazon EC2 console or the command line to find shared AMIs.

Finding a Shared AMI Using the Console

To find a shared private AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. In the first filter, select **Private images**. All AMIs that have been shared with you are listed.

To find a shared public AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. To find shared AMIs, select **Public images** from the **Filter** list.
4. Use filters to list only the types of AMIs that interest you. For example, select **Amazon images** to display only Amazon's public images.

Finding a Shared AMI Using the AWS CLI

To find a shared public AMI using the command line tools

Use the describe-images command to list AMIs. You can scope the list to the types of AMIs that interest you, as shown in the following examples.

The following command lists all public AMIs using the `--executable-users` option. This list includes any public AMIs that you own.

```
$ aws ec2 describe-images --executable-users all
```

The following command lists the AMIs for which you have explicit launch permissions. This list excludes any such AMIs that you own.

```
$ aws ec2 describe-images -executable-users self
```

The following command lists the AMIs owned by Amazon. Amazon's public AMIs have an aliased owner, which appears as `amazon` in the account field. This enables you to find AMIs from Amazon easily. Other users can't alias their AMIs.

```
$ aws ec2 describe-images --owners amazon
```

The following command lists the AMIs owned by the specified AWS account.

```
$ aws ec2 describe-images --owners 123456789012
```

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

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

Finding a Shared AMI Using the Amazon EC2 CLI

To find a shared public AMI using the command line tools

Use the ec2-describe-images command to list AMIs. You can scope the list to the types of AMIs that interest you, as shown in the following examples.

The following command lists all public AMIs using the `-x all` option. This list includes any public AMIs that you own.

```
$ ec2-describe-images -x all
```

The following command lists the AMIs for which you have explicit launch permissions. This list excludes any such AMIs that you own.

```
$ ec2-describe-images -x self
```

The following command lists the AMIs owned by Amazon. Amazon's public AMIs have an aliased owner, which appears as `amazon` in the account field. This enables you to find AMIs from Amazon easily. Other users can't alias their AMIs.

```
$ ec2-describe-images -o amazon
```

The following command lists the AMIs owned by the specified AWS account.

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

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

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

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

Using Shared AMIs

Before you use a shared AMI, take the following steps to confirm that there are no pre-installed credentials that would allow unwanted access to your instance by a third party and no pre-configured remote logging that could transmit sensitive data to a third party. Check the documentation for the Linux distribution used by the AMI for information about improving the security of the system.

To ensure that you don't accidentally lose access to your instance, we recommend that you initiate two SSH sessions and keep the second session open until you've removed credentials that you don't recognize and confirmed that you can still log into your instance using SSH.

1. Identify and disable any unauthorized public SSH keys. The only key in the file should be the key you used to launch the AMI. The following command locates `authorized_keys` files:

```
$ sudo find / -name "authorized_keys" -print -exec cat {} \;
```

2. Disable password-based authentication for the root user. Open the `ssh_config` file and edit the PermitRootLogin line as follows:

```
PermitRootLogin without-password
```

Alternatively, you can disable the ability to log into the instance as root:

```
PermitRootLogin No
```

Then restart the sshd service.

3. Check whether there are any other user accounts that are able to log in to your instance. Accounts with superuser privileges are particularly dangerous. Remove or lock the password of any unknown accounts.

4. Check for open ports that you aren't using and running network services listening for incoming connections.

5. To prevent preconfigured remote logging, you should delete the existing configuration file and restart the rsyslog service. For example:

```
$ sudo rm /etc/rsyslog.config
$ sudo service rsyslog restart
```

6. Verify that all cron jobs are legitimate.

If you discover a public AMI that you feel presents a security risk, contact the AWS security team. For more information, see the AWS Security Center.

Making an AMI Public

Amazon EC2 enables you to share your AMIs with other AWS accounts. You can allow all AWS accounts to launch the AMI (make the AMI public), or only allow a few specific accounts to launch the AMI. You are not billed when your AMI is launched by other AWS accounts; only the accounts launching the AMI are billed.

Before you share an AMI, make sure to read the security considerations in Guidelines for Shared Linux AMIs (p. 66).

> **Note**
> If an AMI has a product code, you can't make it public. You must share the AMI with only specific AWS accounts.

Sharing a Public AMI Using the Console

To share a public AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. Select your AMI in the list, and then select **Modify Image Permissions** from the **Actions** list.
4. Select the **Public** radio button, and then click **Save**.

Sharing a Public AMI Using the AWS CLI

Each AMI has a `launchPermission` property that controls which AWS accounts, besides the owner's, are allowed to use that AMI to launch instances. By modifying the `launchPermission` property of an AMI, you can make the AMI public (which grants launch permissions to all AWS accounts) or share it with only the AWS accounts that you specify.

You can add or remove account IDs from the list of accounts that have launch permissions for an AMI. To make the AMI public, specify the `all` group. You can specify both public and explicit launch permissions.

To make an AMI public

Use the modify-image-attribute command as follows to add the `all` group to the `launchPermission` list for the specified AMI.

```
$ aws ec2 modify-image-attribute --image-id ami-2bb65342 --launch-permission
"{\"Add\":[{\"Group\":\"all\"}]}"
```

To verify the launch permissions of the AMI, use the following describe-image-attribute command.

```
$ aws ec2 describe-image-attribute --image-id ami-2bb65342 --attribute launch
Permission
```

(Optional) To make the AMI private again, remove the `all` group from its launch permissions. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

```
$ aws ec2 modify-image-attribute --image-id ami-2bb65342 "{\"Re
move\":[{\"Group\":\"all\"}]}"
```

Sharing a Public AMI Using the Amazon EC2 CLI

Each AMI has a `launchPermission` property that controls which AWS accounts, besides the owner's, are allowed to use that AMI to launch instances. By modifying the `launchPermission` property of an AMI, you can make the AMI public (which grants launch permissions to all AWS accounts or share it with only the AWS accounts that you specify.

You can add or remove account IDs from the list of accounts that have launch permissions for an AMI To make the AMI public, specify the `all` group. You can specify both public and explicit launch permissions.

To make an AMI public

Use the ec2-modify-image-attribute command as follows to add the `all` group to the `launchPermission` list for the specified AMI.

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

To verify the launch permissions of the AMI, use the following command.

```
$ ec2-describe-image-attribute ami-2bb65342 -l
```

To make the AMI private again, remove the `all` group from its launch permissions. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

```
$ ec2-modify-image-attribute ami-2bb65342 -l -r all
```

Sharing an AMI with Specific AWS Accounts

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

Sharing an AMI Using the Console

To grant explicit launch permissions using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. Select your AMI in the list, and then select **Modify Image Permissions** from the **Actions** list.
4. Specify the AWS account number of the user with whom you want to share the AMI in the **AWS Account Number** field, then click **Add Permission**.

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

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

6. Click **Save** when you are done.

Sharing an AMI Using the AWS CLI

Use the modify-image-attribute command to share an AMI as shown in the following examples.

To grant explicit launch permissions

The following command grants launch permissions for the specified AMI to the specified AWS account.

```
$ aws ec2 modify-image-attribute --image-id ami-2bb65342 --launch-permission
"{\"Add\":[{\"UserId\":\"123456789012\"}]}"
```

To remove launch permissions for an account

The following command removes launch permissions for the specified AMI from the specified AWS account:

```
$ aws ec2 modify-image-attribute --image-id ami-2bb65342 "{\"Re
move\":[{\"UserId\":\"123456789012\"}]}"
```

To remove all launch permissions

The following command removes all public and explicit launch permissions from the specified AMI. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

```
$ aws ec2 reset-image-attribute --image-id ami-2bb65342 --attribute launchPer
mission
```

Sharing an AMI Using the Amazon EC2 CLI

Use the ec2-modify-image-attribute command to share an AMI as shown in the following examples.

To grant explicit launch permissions

The following command grants launch permissions for the specified AMI to the specified AWS account.

```
$ ec2-modify-image-attribute ami-2bb65342 -l -a 111122223333
```

To remove launch permissions for an account

The following command removes launch permissions for the specified AMI from the specified AWS account:

```
$ ec2-modify-image-attribute ami-2bb65342 -l -r 111122223333
```

To remove all launch permissions

The following command removes all public and explicit launch permissions from the specified AMI. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

```
$ ec2-reset-image-attribute ami-2bb65342 -l
```

Using Bookmarks

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

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

To create a bookmark for your AMI

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

```
https://console.aws.amazon.com/ec2/v2/home?region=<region>#LaunchInstanceWiz
ard:ami=<ami_id>
```

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

```
https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWiz
ard:ami=ami-2bb65342
```

2. Distribute the link to users who want to use your AMI.
3. To use a bookmark, click the link or copy and paste it into your browser. The launch wizard opens, with the AMI already selected.

Guidelines for Shared Linux AMIs

If you follow these guidelines, you'll provide a better user experience and make your users' instances less vulnerable to security issues.

Topics
- Update the AMI Tools at Boot Time (p. 67)
- Disable Password-Based Logins for Root (p. 67)
- Disable Root Access (p. 67)
- Remove SSH Host Key Pairs (p. 68)
- Install Public Key Credentials (p. 68)
- Disabling sshd DNS Checks (Optional) (p. 69)
- Identify Yourself (p. 69)
- Protect Yourself (p. 69)

If you are building AMIs for AWS Marketplace, see Building AMIs for AWS Marketplace for guidelines, policies and best practices.

For additional information about sharing AMIs safely, see the following articles:

- How To Share and Use Public AMIs in A Secure Manner
- Public AMI Publishing: Hardening and Clean-up Requirements

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.

For Amazon Linux, add the following to `/etc/rc.local`:

```
# Update the Amazon EC2 AMI tools
echo " + Updating EC2 AMI tools"
yum update -y aws-amitools-ec2
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.

For other distributions, make sure you have the latest AMI tools.

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 disable root access.

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.

Disable Root Access

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 that the root password is not empty. To do this, log into your running instance and issue the following command to disable root access:

```
[ec2-user ~]$ sudo passwd -l root
```

> **Note**
> This command 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

You can securely remove all of these files with the following command.

```
[ec2-user ~]$ sudo shred -u /etc/ssh/*_key /etc/ssh/*_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.

Many distributions, including Amazon Linux and Ubuntu, use the `cloud-init` package to inject public key credentials for a configured user. If your distribution does not support `cloud-init`, you can add the following code to a system start-up script (such as `/etc/rc.local`) to pull in the public key you specified at launch for the `root` user.

```
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 AMI 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 AMI, the shell history contains your secret access key. The following example should be the last
  command executed before bundling from within the instance.

```
[ec2-user ~]$ shred -u ~/.*history
```

AWS recommends using the `--exclude` *directory* option on `ec2-bundle-vol` to skip any directories and subdirectories that contain secret information that you would not like to include in your bundle. For more information, see ec2-bundle-vol in the *Amazon EC2 Command Line Reference*.

- 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 AMIs 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.

Paid AMIs

A *paid AMI* is an AMI that you can purchase from a developer.

Amazon EC2 integrates with AWS Marketplace, enabling developers to charge other Amazon EC2 users for the use of their AMIs or to provide support for instances.

The AWS Marketplace is an online store where you can buy software that runs on AWS; including AMIs that you can use to launch your EC2 instance. The AWS Marketplace AMIs are organized into categories, such as Developer Tools, to enable you to find products to suit your requirements. For more information about AWS Marketplace, see the AWS Marketplace site.

Launching an instance from a paid AMI is the same as launching an instance from any other AMI. No additional parameters are required. The instance is charged according to the rates set by the owner of the AMI, as well as the standard usage fees for the related web services; for example, the hourly rate for running a m1.small instance type in Amazon EC2. The owner of the paid AMI can confirm whether a specific instance was launched using that paid AMI.

Topics

Selling Your AMI

You can sell your AMI using AWS Marketplace. AWS Marketplace offers an organized shopping experience. Additionally, AWS Marketplace also supports AWS features such as Amazon EBS-backed AMIs, Reserved Instances, and Spot Instances.

For information about how to sell your AMI on AWS Marketplace, see Selling on AWS Marketplace.

Finding a Paid AMI

There are several ways that you can find AMIs that are available for you to purchase. For example, you can use AWS Marketplace, the Amazon EC2 console, or the command line. Alternatively, a developer might let you know about a paid AMI themselves.

Finding a Paid AMI Using the Console

To find a paid AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. Select **Public images** from the first **Filter** list, **Marketplace images** from the second **Filter** list, and the operating system from the third **Filter** list.

Finding a Paid AMI Using AWS Marketplace

To find a paid AMI using AWS Marketplace

1. Open AWS Marketplace.
2. Enter the name of the operating system in the search box, and click **Go**.
3. To scope the results further, use one of the categories or filters.
4. Each product is labeled with its product type: either `AMI` or `Software as a Service`.

Finding a Paid AMI Using the AWS CLI

You can find a paid AMI using the describe-images command as follows.

```
$ ec2-describe-images --owners aws-marketplace
```

This command returns numerous details that describe each AMI, including the product code for a paid AMI. The output from `describe-images` includes an entry for the product code like the following:

```
"ProductCodes": [
    {
        "ProductCodeId": "product_code",
        "ProductCodeType": "marketplace"
    }
],
```

Purchase a Paid AMI

You must sign up for (purchase) a paid AMI before you can launch an instance using the AMI.

Typically a seller of a paid AMI presents you with information about the AMI, including its price and a link where you can buy it. When you click the link, you're first asked to log into AWS, and then you can purchase the AMI.

Purchasing a Paid AMI Using the Console

You can purchase a paid AMI by using the Amazon EC2 launch wizard. For more information, see Launching an AWS Marketplace Instance (p. 297).

Subscribing to a Product Using AWS Marketplace

To use the AWS Marketplace, you must have an AWS account. To launch instances from AWS Marketplace products, you must be signed up to use the Amazon EC2 service, and you must be subscribed to the product from which to launch the instance. There are two ways to subscribe to products in the AWS Marketplace:

- **AWS Marketplace website**: You can launch preconfigured software quickly with the 1-Click deployment feature.
- **Amazon EC2 launch wizard**: You can search for an AMI and launch an instance directly from the wizard. For more information, see Launching an AWS Marketplace Instance (p. 297).

Purchasing a Paid AMI From a Developer

The developer of a paid AMI can enable you to purchase a paid AMI that isn't listed in AWS Marketplace. The developer provides you with a link that enables you to purchase the product through Amazon. You can sign in with your Amazon.com credentials and select a credit card that's stored in your Amazon.com account to use when purchasing the AMI.

Getting the Product Code for Your Instance

You can retrieve the AWS Marketplace product code for your instance using its instance metadata. For more information about retrieving metadata, see Instance Metadata and User Data (p. 256).

To retrieve a product code, use the following query:

```
$ GET http://169.254.169.254/latest/meta-data/product-codes
```

If the instance has a product code, Amazon EC2 returns it. For example:

```
774F4FF8
```

Using Paid Support

Amazon EC2 also enables developers to offer support for software (or derived AMIs). Developers can create support products that you can sign up to use. During sign-up for the support product, the developer gives you a product code, which you must then associate with your own AMI. This enables the developer to confirm that your instance is eligible for support. It also ensures that when you run instances of the product, you are charged according to the terms for the product specified by the developer.

> **Important**
> You can't use a support product with Reserved Instances. You always pay the price that's specified by the seller of the support product.

To associate a product code with your AMI, use one of the following commands, where *ami_id* is the ID of the AMI and *product_code* is the product code:

- modify-image-attribute (AWS CLI)

```
$ aws ec2 modify-image-attribute --image-id ami_id --product-codes
"product_code"
```

- ec2-modify-image-attribute (Amazon EC2 CLI)

```
$ ec2-modify-image-attribute ami_id --product-code product_code
```

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

Bills for Paid and Supported AMIs

At the end of each month, you receive an email with the amount your credit card has been charged for using any paid or supported AMIs during the month. This bill is separate from your regular Amazon EC2 bill. For more information, see Paying For AWS Marketplace Products.

Managing Your AWS Marketplace Subscriptions

On the AWS Marketplace website, you can check your subscription details, view the vendor's usage instructions, manage your subscriptions, and more.

To check your subscription details

1. Log in to the AWS Marketplace.
2. Click **Your Account**.
3. Click **Manage Your Software Subscriptions**.
4. All your current subscriptions are listed. Click **Usage Instructions** to view specific instructions for using the product, for example, a user name for connecting to your running instance.

To cancel an AWS Marketplace subscription

1. Ensure that you have terminated any instances running from the subscription.

    a. Open the Amazon EC2 console.
    b. In the navigation pane, click **Instances**.
    c. Select the instance, click **Actions**, and select **Terminate**. When prompted, click **Yes, Terminate**.

2. Log in to the AWS Marketplace, and click **Your Account**, then **Manage Your Software Subscriptions**.
3. Click **Cancel subscription**. You are prompted to confirm your cancellation.

    Note
    After you've canceled your subscription, you are no longer able to launch any instances from that AMI. To use that AMI again, you need to resubscribe to it, either on the AWS Marketplace website, or through the launch wizard in the Amazon EC2 console.

Creating an Amazon EBS-Backed Linux AMI

To create an Amazon EBS-backed Linux AMI, start from an instance that you've launched from an existing Amazon EBS-backed Linux AMI. After you've customized the instance to suit your needs, create and register a new AMI, which you can use to launch new instances with these customizations.

If you need to create an Amazon EBS-backed Windows AMI, see Creating an Amazon EBS-Backed Windows AMI in the *Amazon EC2 User Guide for Microsoft Windows Instances*.

The AMI creation process is different for instance store-backed AMIs. For more information about the differences between Amazon EBS-backed and instance store-backed instances, and how to determine the root device type for your instance, see Storage for the Root Device (p. 55). If you need to create an instance store-backed Linux AMI, see Creating an Instance Store-Backed Linux AMI (p. 77).

Topics
- Overview of the Creation Process for Amazon EBS-Backed AMIs (p. 74)
- Creating the AMI from an Instance (p. 75)
- Creating an AMI from a Snapshot (p. 76)

Overview of the Creation Process for Amazon EBS-Backed AMIs

The following diagram summarizes the creation process for Amazon EBS-backed AMIs.
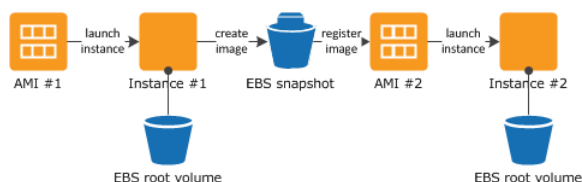
First, launch an instance from an AMI that's similar to the AMI that you'd like to create. You can connect to your instance and customize it. When the instance is set up the way you want it, it's best to stop the instance before you create an AMI to ensure data integrity. Then, you can create the image. When you create an Amazon EBS-backed AMI, we automatically register it for you.

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 tell Amazon EC2 not to power down and reboot the instance. Some file systems, such as xfs, can freeze and unfreeze activity, making it safe to create the image without rebooting the instance.

During the AMI creation process, Amazon EC2 creates snapshots of your instance's root volume and any other Amazon EBS volumes attached to your instance. If any volumes attached to the instance are encrypted, the new AMI will only launch successfully on instances that support Amazon EBS encryption. For more information, see Amazon EBS Encryption (p. 564).

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

Depending on the size of the volumes, it may take several minutes for the entire AMI creation process to complete (sometimes up to 24 hours). To help speed up this process, we recommend that you create

snapshots of your volumes immediately before creating an AMI. For more information, see Creating an Amazon EBS Snapshot (p. 560).

After the process completes, you have a new AMI and snapshot created from the root volume of the instance. When you launch an instance using the new AMI, we create a new Amazon EBS volume for its root volume using the snapshot. Both the AMI and the snapshot incur charges to your account until you delete them. For more information, see Deregistering Your AMI (p. 90).

If you add instance-store volumes or Amazon EBS volumes to your instance in addition to the root device volume, the block device mapping for the new AMI contains information for these volumes, and the block device mappings for instances that you launch from the new AMI automatically contain information for these volumes. The instance-store volumes specified in the block device mapping for the new instance are new and don't contain any data from the instance store volumes of the instance you used to create the AMI. The data on Amazon EBS volumes persists. For more information, see Block Device Mapping (p. 591).

Creating the AMI from an Instance

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

To create an AMI from an instance using the console

1. Open the Amazon EC2 console.
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 (p. 290).
3. (Optional) Connect to the instance and customize it. For example, you can install software and applications, copy data, or attach additional Amazon EBS volumes. For more information about connecting to an instance, see Connect to Your Linux Instance (p. 299).
4. (Optional) Create snapshots of all the volumes attached your instance. This may help reduce the time it takes to create the AMI. For more information about creating snapshots, see Creating an Amazon EBS Snapshot (p. 560).
5. In the navigation pane, click **Instances** and select your instance. Click **Actions**, and then click **Create Image**.

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

6. In the **Create Image** dialog box, specify the following, and then click **Create Image**.

   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.

      > **Warning**
      > If you select **No reboot**, we can't guarantee the file system integrity of the created image.

   d. (Optional) You can modify the root volume, Amazon EBS volumes, and instance store volumes as follows:

      - To change the size of the root volume, locate the **Root** volume in the **Type** column, and fill in the **Size** field.
      - To suppress an Amazon EBS volume specified by the block device mapping of the AMI used to launch the instance, locate the EBS volume in the list and click **Delete**.
      - To add an Amazon EBS volume, click **Add New Volume**, select **EBS** from the **Type** list, and fill in the fields. 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, locate the volume in the list and click **Delete**.
- To add an instance store volume, click **Add New Volume**, select **Instance Store** from the **Type** list, and select a device name from the **Device** list. 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.

7. Click **AMIs** in the navigation pane to view the status of your AMI. While the new AMI is being created, its status is `pending`. This process typically takes a few minutes to finish, and then the status of your AMI is `available`.

8. (Optional) Click **Snapshots** in the navigation pane to view the snapshot that was created for the new AMI. When you launch an instance from this AMI, we use this snapshot to create its root device volume.

To create an AMI from an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- create-image (AWS CLI)
- ec2-create-image (Amazon EC2 CLI)
- New-EC2Image (AWS Tools for Windows PowerShell)

Creating an AMI from a Snapshot

If you have a snapshot of the root device volume of an instance, you can create an AMI from this snapshot using the AWS Management Console or the command line.

To create an AMI from a snapshot using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, under **Elastic Block Store**, click **Snapshots**.
3. Select the snapshot, and then select **Create Image** from the **Actions** list.
4. In the **Create Image from EBS Snapshot** dialog box, do the following:

    a. Select the architecture from the **Architecture** list (`i386` for 32-bit or `x86_64` for 64-bit).

    b. Select the AKI from the **Kernel ID** list. If you select the default AKI or don't select an AKI, you'll be required to specify an AKI every time you launch an instance using this AMI. In addition, your instance may fail the health checks if the default AKI is incompatible with the instance.

    c. Click **Create**.

To create an AMI from a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- register-image (AWS CLI)
- ec2-register (Amazon EC2 CLI)
- Register-EC2Image (AWS Tools for Windows PowerShell)

Creating an Instance Store-Backed Linux AMI

To create an instance store-backed Linux AMI, start from an instance that you've launched from an existing instance store-backed Linux AMI. After you've customized the instance to suit your needs, bundle the volume and register a new AMI, which you can use to launch new instances with these customizations.

If you need to create an instance store-backed Windows AMI, see Creating an Instance Store-Backed Windows AMI in the *Amazon EC2 User Guide for Microsoft Windows Instances*.

The AMI creation process is different for instance store-backed AMIs. For more information about the differences between Amazon EBS-backed and instance store-backed instances, and how to determine the root device type for your instance, see Storage for the Root Device (p. 55). If you need to create an Amazon EBS-backed Linux AMI, see Creating an Amazon EBS-Backed Linux AMI (p. 74).

Topics

Overview of the Creation Process for Instance Store-Backed AMIs

The following diagram summarizes the process of creating an AMI from an instance store-backed instance.
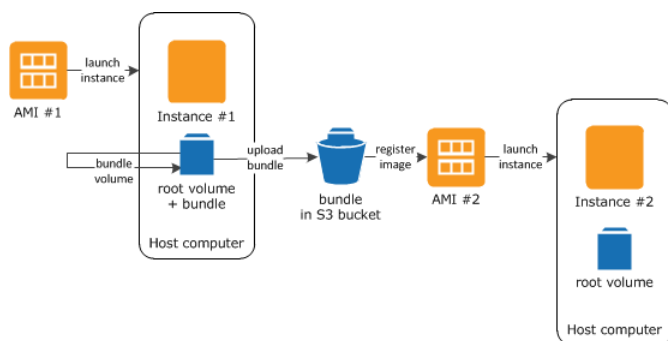
First, launch an instance from an AMI that's similar to the AMI that you'd like to create. You can connect to your instance and customize it. When the instance is set up the way you want it, you can bundle it. It takes several minutes for the bundling process to complete. After the process completes, you have a bundle, which consists of an image manifest (`image.manifest.xml`) and files (`image.part.xx`) that contain a template for the root volume. Next you upload the bundle to your Amazon S3 bucket and then register your AMI.

When you launch an instance using the new AMI, we create the root volume for the instance using the bundle that you uploaded to Amazon S3. The storage space used by the bundle in Amazon S3 incurs charges to your account until you delete it. For more information, see Deregistering Your AMI (p. 90).

If you add instance store volumes to your instance in addition to the root device volume, the block device mapping for the new AMI contains information for these volumes, and the block device mappings for instances that you launch from the new AMI automatically contain information for these volumes. For more information, see Block Device Mapping (p. 591).

Prerequisites

Before you can create the AMI, you must complete the following tasks:

- Install the AMI tools. For more information, see Set Up the AMI Tools.
- Install the API tools. For more information, see Setting Up the Amazon EC2 Command Line Interface Tools on Linux.
- Ensure that you have an Amazon S3 bucket for the bundle. To create an Amazon S3 bucket, open the Amazon S3 console and click **Create Bucket**.

  > **Note**
  > You can also use the AWS CLI mb command to create a bucket. To get started with the AWS CLI, see AWS Command Line Interface User Guide.

- Ensure that you have the following credentials:
  - Your AWS account ID. To retrieve your account ID, go to Your Security Credentials and expand **Account Identifiers**.
  - An X.509 certificate and private key. If you need to create an X.509 certificate, go to Your Security Credentials, expand **X.509 Certificates**, and click **Create New Certificate**. The X.509 certificate and private key are used to encrypt and decrypt your AMI.
  - Your access key ID. If you need to retrieve or create an access key ID, go to Your Security Credentials, and expand **Access Keys**.
  - Your secret access key. You can't retrieve your secret access key. Therefore, if you can't find your secret access key, you'll need to create a new one. To create a secret access key, go to Your Security Credentials, expand **Access Keys**, and click **Create New Access Key**.
- Connect to your instance and customize it. For example, you can install software and applications, copy data, delete temporary files, and modify the Linux configuration.

Creating an AMI from an Instance Store-Backed Linux Instance

To prepare to use the Amazon EC2 AMI Tools (HVM instances only)

The Amazon EC2 AMI tools require GRUB Legacy to boot properly. Some AMIs (most notably, Ubuntu) are configured to use GRUB 2. You must check to see that your instance uses GRUB Legacy, and if not, you need to install and configure it.

HVM instances also require partitioning tools to be installed for the AMI tools to work properly.

1. Check to see if your instance uses GRUB Legacy.

   a. List the block devices to find the root block device.

   ```
   [ec2-user ~]$ lsblk
   NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
   xvda    202:0    0   8G  0 disk
     xvda1 202:1    0   8G  0 part /
   xvdb    202:16   0  30G  0 disk /media/ephemeral0
   ```

In this example, the root device (indicated by a `MOUNTPOINT` of /) is `/dev/xvda1`. The root block device is its parent, `/dev/xvda`.

b. Determine the GRUB version on the root block device.

```
[ec2-user ~]$ sudo file -s /dev/xvda
/dev/xvda: x86 boot sector; GRand Unified Bootloader, stage1 version
0x3, stage2 address 0x2000, 1st sector stage2 0x800, stage2 segment
0x200, GRUB version 0.94; partition 1: ID=0xee, starthead 254,
startsector 1, 16777215 sectors, extended partition table (last)\011,
code offset 0x48
```

In the above example, the GRUB version is 0.94, which is GRUB Legacy. If your GRUB version is 0.9*x* or less, you may move on to Step 3 (p. 79). If you do not see a GRUB version in this output, try the **grub-install -v** command.

```
ubuntu:~$ grub-install -v
grub-install (GRUB) 1.99-21ubuntu3.10
```

In this example, the GRUB version is greater than 0.9*x*, so GRUB Legacy must be installed. Proceed to Step 2 (p. 79).

2. Install the `grub` package using the package manager for your distribution to install GRUB Legacy. For Ubuntu instances, use the following command.

```
ubuntu:~$ sudo apt-get install -y grub
```

You can verify that your instance is using GRUB Legacy with the **grub --version** command.

```
ubuntu:~$ grub --version
grub (GNU GRUB 0.97)
```

3. Install the following partition management packages using the package manager for your distribution.

- `gdisk` (some distributions may call this package `gptfdisk` instead)
- `kpartx`

For Ubuntu instances, use the following command.

```
ubuntu:~$ sudo apt-get install -y gdisk kpartx
```

4. Check the kernel parameters for your instance.

```
ubuntu:~$ cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-3.2.0-54-virtual root=UUID=4f392932-ed93-4f8f-aee7-
72bc5bb6ca9d ro console=ttyS0 xen_emul_unplug=unnecessary
```

Note the options following the kernel and root device parameters, `console=ttyS0` and `xen_emul_unplug=unnecessary`.

5. Check the kernel entries in `/boot/grub/menu.lst`.

```
ubuntu:~$ grep ^kernel /boot/grub/menu.lst
kernel  /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro con
sole=hvc0
kernel  /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro single
kernel  /boot/memtest86+.bin
```

Note that the `console` parameter is pointing to `hvc0` instead of `ttyS0` and that the `xen_emul_un-plug=unnecessary` parameter is missing.

6. Edit the `/boot/grub/menu.lst` file with your favorite text editor (such as **vim** or **nano**) to change the console and add the parameters you identified earlier to the boot entries.

```
title          Ubuntu 12.04.3 LTS, kernel 3.2.0-54-virtual
root           (hd0)
kernel         /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs
ro console=ttyS0 xen_emul_unplug=unnecessary
initrd         /boot/initrd.img-3.2.0-54-virtual

title          Ubuntu 12.04.3 LTS, kernel 3.2.0-54-virtual (recovery mode)
root           (hd0)
kernel         /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs
ro single console=ttyS0 xen_emul_unplug=unnecessary
initrd         /boot/initrd.img-3.2.0-54-virtual

title          Ubuntu 12.04.3 LTS, memtest86+
root           (hd0)
kernel         /boot/memtest86+.bin
```

7. Verify that your kernel entries now contain the correct parameters.

```
ubuntu:~$ grep ^kernel /boot/grub/menu.lst
kernel  /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro con
sole=ttyS0 xen_emul_unplug=unnecessary
kernel  /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro single
 console=ttyS0 xen_emul_unplug=unnecessary
kernel  /boot/memtest86+.bin
```

To create an AMI from an instance store-backed Linux instance

This procedure assumes that you have satisfied the prerequisites in Prerequisites (p. 78).

1. Upload your credentials to your instance. We use these credentials to ensure that only you and Amazon EC2 can access your AMI.

    a. Create a temporary directory on your instance for your credentials as follows:

    ```
    [ec2-user ~]$ mkdir /tmp/cert
    ```

    This enables you to exclude your credentials from the created image.

    b. Copy your X.509 certificate and private key from your computer to the `/tmp/cert` directory on your instance, using a secure copy tool such as scp (p. 301). The `-i` *my-private-key*.pem option in the following **scp** command is the private key you use to connect to your instance with SSH, not the X.509 private key. For example:

```
you@your_computer:~ $ scp -i my-private-key.pem /path/to/pk-HKZYK
TAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem /path/to/cert-HKZYKTAIG2ECMXY
IBH3HXV4ZBEXAMPLE.pem ec2-user@ec2-203-0-113-25.compute-1.amazon
aws.com:/tmp/cert/
pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem  100%  717     0.7KB/s   00:00
cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem  100%  685     0.7KB/s   00:00
```

2. Prepare the bundle to upload to Amazon S3 using the ec2-bundle-vol command. Be sure to specify
    the `-e` option to exclude the directory where your credentials are stored. By default, the bundle process
    excludes files that might contain sensitive information. These files include `*.sw`, `*.swo`, `*.swp`,
    `*.pem`, `*.priv`, `*id_rsa*`, `*id_dsa*` `*.gpg`, `*.jks`, `*/.ssh/authorized_keys`, and
    `*/.bash_history`. To include all of these files, use the `--no-filter` option. To include some of
    these files, use the `--include` option.

    > **Important**
    > By default, the AMI bundling process creates a compressed, encrypted collection of files in
    > the `/tmp` directory that represent your root volume. If you do not have enough free disk
    > space in `/tmp` to store the bundle, you need to specify a different location for the bundle to
    > be stored with the `-d /path/to/bundle/storage` option. Some instances have ephem-
    > eral storage mounted at `/mnt` or `/media/ephemeral0` that you can use, or you can also
    > create (p. 526), attach (p. 529), and mount (p. 531) a new Amazon EBS volume to store the
    > bundle.

    a. The **ec2-bundle-vol** command needs to run as `root`. For most commands, you can use **sudo**
        to gain elevated permissions, but in this case, you should run **sudo -E su** to keep your environ-
        ment variables.

        ```
        [ec2-user ~]$ sudo -E su
        ```

    b. Run the ec2-bundle-vol command with the following arguments. If you do not have enough
        available disk space in `/tmp` to store your bundle, specify a location that has available space
        with the `-d /path/to/bundle/storage` option. For HVM instances, be sure to add the
        `--partition` flag; otherwise, your AMI will not boot. For more information on this command
        and its available options, see ec2-bundle-vol in the Amazon EC2 Command Line Reference.

        ```
        [root ec2-user]# $EC2_AMITOOL_HOME/bin/ec2-bundle-vol -k /tmp/cert/pk-
        HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c /tmp/cert/cert-HKZYKTAIG2ECMXY
        IBH3HXV4ZBEXAMPLE.pem -u your_aws_account_id -r x86_64 -e /tmp/cert
        ```

        It can take a few minutes to create the image. When this command completes, your `tmp` directory
        contains the bundle (`image.manifest.xml`, plus multiple `image.part.xx` files).

    c. Exit from the `root` shell.

        ```
        [root ec2-user]# exit
        ```

3. Upload your bundle to Amazon S3 using the ec2-upload-bundle command. Note that if the bundle
    prefixes (directories) don't exist in the bucket, this command creates them.

    > **Note**
    > If you specified a path with the `-d /path/to/bundle/storage` option in Step 2.b (p. 81),
    > use that same path in the `-m` option below, instead of `/tmp`.

```
[ec2-user ~]$ ec2-upload-bundle -b my-s3-bucket/bundle_folder/bundle_name
-m /tmp/image.manifest.xml -a your_access_key_id -s your_secret_access_key
 --region us-west-2
```

4. (Optional) After the bundle is uploaded to Amazon S3, you can remove the bundle from the `/tmp` directory on the instance using the following **rm** command:

    > **Note**
    > If you specified a path with the `-d /path/to/bundle/storage` option in Step 2.b (p. 81), use that same path below, instead of `/tmp`.

    ```
    [ec2-user ~]$ sudo rm /tmp/image.manifest.xml /tmp/image.part.* /tmp/image
    ```

5. Register your AMI using the ec2-register command. Note that you don't need to specify the `-O` and `-W` options if you've set the `AWS_ACCESS_KEY` and `AWS_SECRET_KEY` environment variables.

    > **Important**
    > For HVM AMIs, add the `--virtualization-type hvm` flag.

    ```
    [ec2-user ~]$ ec2-register my-s3-bucket/bundle_folder/bundle_name/image.mani
    fest.xml -n AMI_name -O your_access_key_id -W your_secret_access_key --region
     us-west-2
    ```

Converting your Instance Store-Backed AMI to an Amazon EBS-Backed AMI

You can convert an instance store-backed Linux AMI that you own to an Amazon EBS-backed Linux AMI.

> **Important**
> You can't convert an instance store-backed Windows AMI to an Amazon EBS-backed Windows AMI and you cannot convert an AMI that you do not own.

To convert an instance store-backed AMI to an Amazon EBS-backed AMI

1. Launch an Amazon Linux instance from an Amazon EBS-backed AMI. For more information, see Launching an Instance (p. 290). Amazon Linux instances have the Amazon EC2 command line and AMI tools pre-installed.

2. Upload the X.509 private key that you used to bundle your instance store-backed AMI to your instance. We use this key to ensure that only you and Amazon EC2 can access your AMI.

    a. Create a temporary directory on your instance for your X.509 private key as follows:

        ```
        [ec2-user ~]$ mkdir /tmp/cert
        ```

    b. Copy your X.509 private key from your computer to the `/tmp/cert` directory on your instance, using a secure copy tool such as scp (p. 301). The `my-private-key` parameter in the following command is the private key you use to connect to your instance with SSH. For example:

        ```
        you@your_computer:~ $ scp -i my-private-key.pem /path/to/pk-HKZYK
        TAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-user@ec2-203-0-113-25.compute-
        ```

```
1.amazonaws.com:/tmp/cert/
pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem  100%   717     0.7KB/s   00:00
```

3. Set environment variables for your AWS access key and secret key.

```
[ec2-user ~]$ export AWS_ACCESS_KEY=your_access_key_id
[ec2-user ~]$ export AWS_SECRET_KEY=your_secret_access_key
```

4. Prepare an Amazon EBS volume for your new AMI.

    a. Create an empty Amazon EBS volume in the same Availability Zone as your instance using the
        ec2-create-volume command. Note the volume ID in the command output.

        Important
        This Amazon EBS volume must be the same size or larger than the original instance
        store root volume.

```
[ec2-user ~]$ ec2-create-volume --size 10 --region us-west-2 --availab
ility-zone us-west-2b
VOLUME volume_id 10  us-west-2b creating 2014-01-24T23:11:45+0000
standard
```

    b. Attach the volume to your Amazon EBS-backed instance using the ec2-attach-volume command.

```
[ec2-user ~]$ ec2-attach-volume volume_id -i instance_id --device /dev/sdb
 --region us-west-2
ATTACHMENT volume_id instance_id /dev/sdb attaching 2014-01-
24T23:15:34+0000
```

5. Create a folder for your bundle.

```
[ec2-user ~]$ mkdir /tmp/bundle
```

6. Download the bundle for your instance store-based AMI to `/tmp/bundle` using the ec2-download-
    bundle command.

```
[ec2-user ~]$ ec2-download-bundle -b my-s3-bucket/bundle_folder/bundle_name
 -m image.manifest.xml -a $AWS_ACCESS_KEY -s $AWS_SECRET_KEY --privatekey
/path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -d /tmp/bundle
```

7. Reconstitute the image file from the bundle using the ec2-unbundle command.

    a. Change directories to the bundle folder.

```
[ec2-user ~]$ cd /tmp/bundle/
```

    b. Run the ec2-unbundle command.

```
[ec2-user bundle]$ ec2-unbundle -m image.manifest.xml --privatekey
/path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
```

8. Copy the files from the unbundled image to the new Amazon EBS volume.

```
[ec2-user bundle]$ sudo dd if=/tmp/bundle/image of=/dev/sdb bs=1M
```

9. Create a mount point for the new Amazon EBS volume and mount the volume.

```
[ec2-user bundle]$ sudo mkdir /mnt/ebs
[ec2-user bundle]$ sudo mount /dev/sdb /mnt/ebs
```

10. Open the /etc/fstab file on the EBS volume with your favorite text editor (such as **vim** or **nano**) and remove any entries for instance store (ephemeral) volumes. Because the Amazon EBS volume is mounted on /mnt/ebs, the fstab file is located at /mnt/ebs/etc/fstab.

```
[ec2-user bundle]$ sudo nano /mnt/ebs/etc/fstab
#
LABEL=/      /         ext4    defaults,noatime  1   1
tmpfs       /dev/shm   tmpfs   defaults          0   0
devpts      /dev/pts   devpts  gid=5,mode=620    0   0
sysfs       /sys       sysfs   defaults          0   0
proc        /proc      proc    defaults          0   0
/dev/sdb        /media/ephemeral0      auto    defaults,comment=cloudconfig
    0       2
```

In the above example, the last line should be removed.

11. Unmount the volume and detach it from the instance.

```
[ec2-user bundle]$ sudo umount /mnt/ebs
[ec2-user bundle]$ ec2-detach-volume volume_id --region us-west-2
ATTACHMENT volume_id instance_id /dev/sdb detaching 2014-01-24T23:15:34+0000
```

12. Create an AMI from the new Amazon EBS volume as follows.

   a. Create a snapshot of the new Amazon EBS volume.

```
[ec2-user bundle]$ ec2-create-snapshot --region us-west-2 -d
"your_snapshot_description" -O $AWS_ACCESS_KEY -W $AWS_SECRET_KEY
volume_id
SNAPSHOT snapshot_id volume_id pending 2014-01-25T00:18:48+000
1234567891011 10 your_snapshot_description
```

   b. Check to see that your snapshot is complete.

```
[ec2-user bundle]$ ec2-describe-snapshots --region us-west-2 snapshot_id
SNAPSHOT snapshot_id volume_id completed 2014-01-25T00:18:48+0000 100%
1234567891011 10 your_snapshot_description
```

c.   Identify the architecture and the kernel image (`aki`) used on the original AMI with the **ec2-de-scribe-images** command. You need the AMI ID of the original instance store-backed AMI for this step.

```
[ec2-user bundle]$ ec2-describe-images --region us-west-2 ami-id
IMAGE ami-8ef297be amazon/amzn-ami-pv-2013.09.2.x86_64-s3 amazon available
 public  x86_64 machine aki-fc8f11cc   instance-store paravirtual xen
```

In this example, the architecture is `x86_64` and the kernel image ID is `aki-fc8f11cc`. Use these values in the following step. If the output of the above command also lists an `ari` ID, take note of that as well.

d.   Register your new AMI with the snapshot ID of your new Amazon EBS volume and the values from the previous step. If the previous command output listed an `ari` ID, include that in the following command with `--ramdisk ari_id`.

```
[ec2-user bundle]$ ec2-register --region us-west-2 -n your_new_ami_name
 -s snapshot_id -a x86_64 --kernel aki-fc8f11cc
IMAGE new-ami-id
```

13. (Optional) After you have tested that you can launch an instance from your new AMI, you can delete the Amazon EBS volume that you created for this procedure.

```
$ ec2-delete-volume volume_id
```

Copying an AMI

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

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

Copying your AMIs provides the following benefits:

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

AMI Copy

You can copy both Amazon EBS-backed AMIs and instance-store-backed AMIs. You can copy an AMI to as many regions as you like. You can also copy an AMI to the same region. Each copy of an AMI results

in a new AMI with its own unique AMI ID. When you launch an instance from an AMI, we launch it into the same region as the AMI you select, as shown in the following diagram.
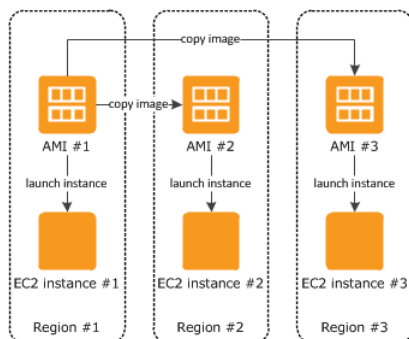
When you copy an AMI, the new AMI is fully independent of the source AMI; there is no link to the original (source) AMI. You can modify the new AMI without affecting the source AMI. The reverse is also true: you can modify the source AMI without affecting the new AMI. Therefore, if you make changes to the source AMI and want those changes to be reflected in the AMI in the destination region, you must recopy the source AMI to the destination region.

We don't copy launch permissions, user-defined tags, or Amazon S3 bucket permissions from the source AMI to the new AMI. After the copy operation is complete, you can apply launch permissions, user-defined tags, and Amazon S3 bucket permissions to the new AMI. AMIs with encrypted volumes cannot be copied.

We try to find matching AKIs and ARIs for the new AMI in the destination region. If we can't find a matching AKI or ARI, then we don't copy the AMI. If you are using the AKIs and ARIs that we recommend, the copy operation registers the AMI with the appropriate AKI and ARI in the destination region. If you get an error message "Failed to find matching AKI/ARI", it means that the destination region doesn't contain an AKI or ARI that matches those specified in the source AMI. If your AMI uses 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 PV-GRUB (p. 98).

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

Copying an Amazon EC2 AMI

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

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

To copy an AMI using the console

1. Open the Amazon EC2 console.
2. From the navigation bar, select the region that contains the AMI to copy.
3. In the navigation pane, click **AMIs**.
4. Select the AMI to copy, click **Actions**, and then click **Copy AMI**. If your AMI has encrypted volumes, see Copying an Amazon EC2 AMI with Encrypted Volumes (p. 87).

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

5.   In the **AMI Copy** page, set the following fields, and then click **Copy AMI**:

  • **Destination region**: Select the region to which you want to copy the AMI.
  • **Name**: Specify a name for the new AMI.
  • **Description**: By default, the description includes information about the source AMI so that you can identify a copy from the original. You can change this description as necessary.

6.   We display a confirmation page to let you know that the copy operation has been initiated and provide you with the ID of the new AMI.

  To check on the progress of the copy operation immediately, click the provided link to switch to the destination region. To check on the progress later, click **Done**, and then when you are ready, use the navigation pane to switch to the destination region.

  The initial status of the destination AMI is `pending` and the operation is complete when the status is `available`.

To copy an AMI using the command line

Copying an AMI from the command line requires that you specify both the source and destination regions. You specify the source region using the `--source-region` parameter. For the destination region, you have two options:

• Use the `--region` parameter.
• Set an environmental variable. For more information, see Setting up the CLI Tools (Linux).

You can copy an AMI using one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

• copy-image (AWS CLI)
• ec2-copy-image (Amazon EC2 CLI)
• Copy-EC2Image (AWS Tools for Windows PowerShell)

Copying an Amazon EC2 AMI with Encrypted Volumes

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

To copy an Amazon EC2AMI with encrypted volumes using the AWS CLI

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

  • `VirtualizationType`
  • `SriovNetSupport`
  • `BlockDeviceMappings`, including the `SnapshotId`, `VolumeType`, and `Encrypted` values
  • `Architecture`

- RootDeviceName

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

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

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

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

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

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

Stopping a Pending AMI Copy Operation

You can stop a pending AMI copy using the AWS Management Console or the command line.

To stop an AMI copy operation using the console

1. Open the Amazon EC2 console.
2. From the navigation bar, select the destination region from the region selector.
3. In the navigation pane, click **AMIs**.
4. Select the AMI you want to stop copying, click **Actions**, and then click **Deregister**.
5. When asked for confirmation, click **Continue**.

To stop an AMI copy operation using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- deregister-image (AWS CLI)
- ec2-deregister (Amazon EC2 CLI)
- Unregister-EC2Image (AWS Tools for Windows PowerShell)

Deregistering Your AMI

You can deregister an AMI when you have finished using it. After you deregister an AMI, you can't use it to launch new instances.

When you deregister an AMI, it doesn't affect any instances that you've already launched from the AMI. You'll continue to incur usage costs for these instances. Therefore, if you are finished with these instances, you should terminate them.

The procedure that you'll use to clean up your AMI depends on whether it is backed by Amazon EBS or instance store.

Topics

- Cleaning Up Your Amazon EBS-Backed AMI (p. 90)
- Cleaning Up Your Instance Store-Backed AMI (p. 91)

Cleaning Up Your Amazon EBS-Backed AMI

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

The following diagram illustrates the process for cleaning up your Amazon EBS-backed AMI.
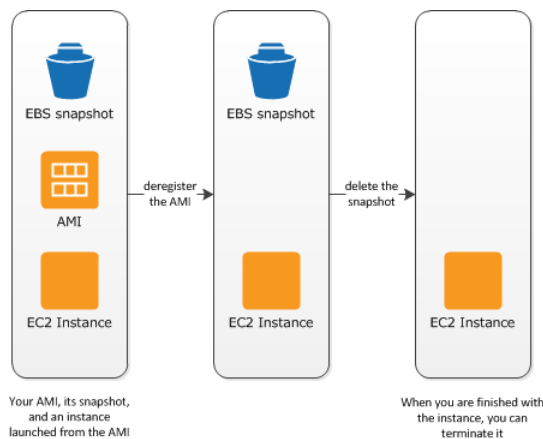
To clean up your Amazon EBS-backed AMI

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**. Select the AMI, click **Actions**, and then click **Deregister**. When prompted for confirmation, click **Continue**.

   The AMI status is now `unavailable`.
3. In the navigation pane, click **Snapshots**. Select the snapshot and click **Delete Snapshot**. When prompted for confirmation, click **Yes, Delete**.
4. (Optional) If you are finished with an instance that you launched from the AMI, terminate it. In the navigation pane, click **Instances**. Select the instance, click **Actions**, and then click **Terminate.** When prompted for confirmation, click **Yes, Terminate**.

Cleaning Up Your Instance Store-Backed AMI

When you deregister an instance store-backed AMI, it doesn't affect the files that you uploaded to Amazon S3 when you created the AMI. You'll continue to incur usage costs for these files in Amazon S3. Therefore, if you are finished with these files, you should delete them.

The following diagram illustrates the process for cleaning up your instance store-backed AMI.
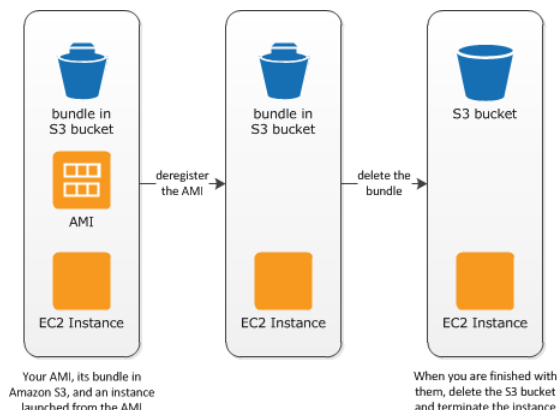
To clean up your instance store-backed AMI

1. Deregister the AMI using the ec2-deregister command as follows.

    ```
    ec2-deregister ami_id
    ```

    The AMI status is now unavailable.

2. Delete the bundle using the ec2-delete-bundle command as follows.

    ```
    ec2-delete-bundle -b myawsbucket/myami -a your_access_key_id -s
    your_secret_access_key -p image
    ```

3. (Optional) If you are finished with an instance that you launched from the AMI, you can terminate it using the ec2-terminate-instances command as follows.

    ```
    ec2-terminate-instances instance_id
    ```

4. (Optional) If you are finished with the Amazon S3 bucket that you uploaded the bundle to, you can delete the bucket. To delete an Amazon S3 bucket, open the Amazon S3 console, select the bucket, click **Actions**, and then click **Delete**.

Amazon Linux

Amazon Linux is provided by Amazon Web Services (AWS). 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. AWS provides ongoing security and maintenance updates to all instances running Amazon Linux.

To launch an Amazon Linux instance, use an Amazon Linux AMI. AWS provides Amazon Linux AMIs to Amazon EC2 users at no additional cost.

Topics

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.

Amazon Linux does not allow remote root SSH by default. Also, password authentication is disabled to prevent brute-force password attacks. To enable SSH logins to an Amazon Linux instance, 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 using SSH is `ec2-user`; this account also 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 about launching and using your Amazon Linux instance, see Launch Your Instance (p. 289). For information about connecting to your Amazon Linux instance, see Connecting to Your Linux Instance (p. 300).

Identifying Amazon Linux AMI Images

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

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

```
[ec2-user ~]$ cat /etc/image-id
image_name="amzn-ami-hvm"
image_version="2014.09"
image_arch="x86_64"
```

```
image_file="amzn-ami-hvm-2014.09.0.x86_64.ext4.gpt"
image_stamp="0136-7efc"
image_date="20140918210050"
recipe_name="amzn ami"
recipe_id="77f9d528-2175-cf83-7010-5542-dd61-7787-34df9a92"
```

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 Amazon Linux. This file will not change as you install updates from the **yum** repository.

Amazon Linux contains an `/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.

The following is an example of an `/etc/system-release` file:

```
[ec2-user ~]$ cat /etc/system-release
Amazon Linux AMI release 2014.09
```

Amazon Linux 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 Amazon Linux or in the default 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`

Although the `aws-apitools-*` command line tools are included with every Amazon Linux version, the `aws-cli` command line tools provide a standard experience across all Amazon Web Services and will eventually replace the service-specific tool sets.

For instances launched using IAM roles, a simple script has been included to prepare `AWS_CREDEN-TIAL_FILE`, `JAVA_HOME`, `AWS_PATH`, `PATH`, and product-specific environment variables after a credential file has been installed to simplify the configuration of these tools.

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

/opt/aws/bin
> Symbolic links 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 a symbolic link *name* that is 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, such as EC2_HOME.

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. Amazon Linux 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. Amazon Linux also uses cloud-init to perform initial configuration of the ec2-user account.

For more information about cloud-init, see http://cloudinit.readthedocs.org/en/latest/.

Amazon Linux uses the following cloud-init actions (configurable in /etc/sysconfig/cloudinit):

- action: INIT (always runs)
  - Sets a default locale
  - Sets the hostname
  - Parses and handles user data
- action: CONFIG_SSH
  - Generates host private SSH keys
  - Adds a user's public SSH keys to .ssh/authorized_keys for easy login and administration
- action: PACKAGE_SETUP
  - Prepares **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 configuration 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 decompresses the data and handles it appropriately.
- MIME multipart

- Using a MIME 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 decodes the data and handles it appropriately. If not, it returns the base64 data intact.
- User-Data script
  - Begins with `#!` or `Content-Type: text/x-shellscript`.
  - The script is executed by `/etc/init.d/cloud-init-user-scripts` during the first boot cycle. This occurs late in the boot process (after the initial configuration actions are 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 is read, and their content 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 configuration 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 one time. 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 Amazon Linux, 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 Amazon Linux to the next. For example, if you launch an instance from an older version of the Amazon Linux AMI (such as 2014.03 or earlier) and run `yum update -y`, you end up with the latest packages.

You can disable rolling updates for Amazon Linux 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 2014.03 AMI and have it receive only the updates that were released prior to the 2014.09 AMI, until you are ready to migrate to the 2014.09 AMI. To enable lock-on-launch in new instances, launch it with the following user data passed to `cloud-init`, using either the Amazon EC2 console or the `ec2-run-instances` command with the `-f` flag.

```
#cloud-config
repo_releasever: 2014.03
```

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

Amazon Linux 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 using **yum** update tools, as well as on the Amazon Linux AMI packages site. Hosting repositories in each region enables us to deploy updates quickly and without any data transfer charges. The packages can be installed by issuing **yum** commands, such as the following example:

```
[ec2-user ~]$ 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 repositories. The third-party packages are not supported by AWS.

If you find that Amazon Linux does not contain an application you need, you can simply install the application directly on your Amazon Linux instance. Amazon Linux uses RPMs 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 repository first, because many applications are available there. These applications can easily be added to your Amazon Linux instance.

To upload your applications onto a running Amazon Linux 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 the built-in `cloud-init` package. For more information, see `cloud-init` (p. 94).

> **Important**
> If your instance is running in a virtual private cloud (VPC), you must attach an Internet Gateway to the VPC in order to contact the **yum** repository. For more information, see Internet Gateways in the *Amazon VPC User Guide*.

Accessing Source Packages for Reference

You can view the source of packages you have installed on your instance for reference purposes by using tools provided in Amazon Linux. Source packages are available for all of the packages included in Amazon Linux 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:

```
[ec2-user ~]$ get_reference_source -p bash
```

The following is a sample response:

```
Requested package: bash
Found package from local RPM database: bash-4.1.2-15.17.amzn1.x86_64
Corresponding source RPM to found package :
bash-4.1.2-15.17.amzn1.src.rpm

Are these parameters correct? Please type 'yes' to continue: yes
Source RPM downloaded to:
/usr/src/srpm/debug/bash-4.1.2-15.17.amzn1.src.rpm
```

The source RPM is placed in the `/usr/src/srpm/debug` directory of your 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 is available for use.

> **Important**
> If your instance is running in a virtual private cloud (VPC), you must attach an Internet Gateway to the VPC in order to contact the **yum** repository. For more information, see Internet Gateways in the *Amazon VPC User Guide*.

Developing Applications

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

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 Amazon Linux instances, you can simply relaunch new instances with the latest Amazon Linux AMI. If you need to preserve data or customizations for your Amazon Linux instances, you can maintain those instances through the Amazon Linux **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 use, even as new versions are released. In some cases, if you're seeking support for an older version of Amazon Linux; through AWS Support, we might ask you to move to newer versions as part of the support process.

Security Updates

Security updates are provided via the Amazon Linux AMI **yum** repositories as well as via updated Amazon Linux AMIs. Security alerts are published in the Amazon Linux AMI Security Center. For more information on AWS security policies or to report a security problem, go to 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 performs the security upgrades at launch for any packages installed at that time. The Amazon Linux AMI also notifies you of any updates to the installed packages by listing the number of available updates upon login using the `/etc/motd` file. To install these updates, you need to run **sudo yum upgrade** on the instance.

> **Important**
> If your instance is running in a virtual private cloud (VPC), you must attach an Internet Gateway to the VPC in order to contact the **yum** repository. For more information, see Internet Gateways in the *Amazon VPC User Guide*.

Support

Support for installation and use of the base Amazon Linux AMI is included through subscriptions to AWS Support. For more information, see AWS Support.

We encourage you to post any questions you have about Amazon Linux to the Amazon EC2 forum.

PV-GRUB

Amazon Machine Images that use paravirtual (PV) virtualization use a system called *PV-GRUB* during the boot process. PV-GRUB is a paravirtual boot loader that runs a patched version of GNU GRUB 0.97. When you start an instance, PV-GRUB starts the boot process and then chain loads the kernel specified by your image's `menu.lst` file.

PV-GRUB understands standard `grub.conf` or `menu.lst` commands, which allows it to work with all currently supported Linux distributions. Older distributions such as Ubuntu 10.04 LTS, Oracle Enterprise Linux or CentOS 5.x require a special "ec2" or "xen" kernel package, while newer distributions include the required drivers in the default kernel package.

Most modern paravirtual AMIs use a PV-GRUB AKI by default (including all of the paravirtual Linux AMIs available in the Amazon EC2 Launch Wizard Quick Start menu), so there are no additional steps that you need to take to use a different kernel on your instance, provided that the kernel you want to use is compatible with your distribution. The best way to run a custom kernel on your instance is to start with an AMI that is close to what you want and then to compile the custom kernel on your instance and modify the `menu.lst` file as shown in Configuring GRUB (p. 99) to boot with that kernel.

You can verify that the kernel image for an AMI is a PV-GRUB AKI by executing the following command with the Amazon EC2 command line tools (substituting the kernel image ID you want to check):

```
$ ec2-describe-images -a -F image-id=aki-880531cd
IMAGE    aki-880531cd    amazon/pv-grub-hd0_1.04-x86_64.gz ...
```

The `name` field of the output should contain `pv-grub`.

Topics
- Limitations of PV-GRUB (p. 99)
- Configuring GRUB (p. 99)
- Amazon PV-GRUB Kernel Image IDs (p. 100)

- Updating PV-GRUB (p. 102)

Limitations of PV-GRUB

PV-GRUB has the following limitations:

- You can't use the 64-bit version of PV-GRUB to start a 32-bit kernel or vice versa.
- You can't specify an Amazon ramdisk image (ARI) when using a PV-GRUB AKI.
- AWS has tested and verified that PV-GRUB works with these file system formats: EXT2, EXT3, EXT4, JFS, XFS, and ReiserFS. Other file system formats might not work.
- PV-GRUB can boot kernels compressed using the gzip, bzip2, lzo, and xz compression formats.
- Cluster AMIs don't support or need PV-GRUB, because they use full hardware virtualization (HVM). While paravirtual instances use PV-GRUB to boot, HVM instance volumes are treated like actual disks, and the boot process is similar to the boot process of a bare metal operating system with a partitioned disk and bootloader.
- PV-GRUB versions 1.03 and earlier don't support GPT partitioning; they support MBR partitioning only.
- 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.

Configuring GRUB

To boot PV-GRUB, a GRUB `menu.lst` file must exist in the image; the most common location for this file is `/boot/grub/menu.lst`.

The following is an example of a `menu.lst` configuration file for booting an AMI with a PV-GRUB AKI. In this example, there are two kernel entries to choose from: `Amazon Linux 2013.09` (the original kernel for this AMI), and `Vanilla Linux 3.11.6` (a newer version of the Vanilla Linux kernel from https://www.kernel.org/). The Vanilla entry was copied from the original entry for this AMI, and the `kernel` and `initrd` paths were updated to the new locations. The `default 0` parameter points the boot loader to the first entry it sees (in this case, the Vanilla entry), and the `fallback 1` parameter points the bootloader to the next entry if there is a problem booting the first.

```
default 0
fallback 1
timeout 0
hiddenmenu

title Vanilla Linux 3.11.6
root (hd0)
kernel /boot/vmlinuz-3.11.6 root=LABEL=/ console=hvc0
initrd /boot/initrd.img-3.11.6

title Amazon Linux 2013.09 (3.4.62-53.42.amzn1.x86_64)
root (hd0)
kernel /boot/vmlinuz-3.4.62-53.42.amzn1.x86_64 root=LABEL=/ console=hvc0
initrd /boot/initramfs-3.4.62-53.42.amzn1.x86_64.img
```

You don't need to specify a fallback kernel in your `menu.lst` file, but we recommend that you have a fallback when you test a new kernel. PV-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 isn't found.

PV-GRUB checks the following locations for `menu.lst`, using the first one it finds:

- `(hd0)/boot/grub`
- `(hd0,0)/boot/grub`
- `(hd0,0)/grub`
- `(hd0,1)/boot/grub`
- `(hd0,1)/grub`
- `(hd0,2)/boot/grub`
- `(hd0,2)/grub`
- `(hd0,3)/boot/grub`
- `(hd0,3)/grub`

Note that PV-GRUB 1.03 and earlier only check one of the first two locations in this list.

Amazon PV-GRUB Kernel Image IDs

PV-GRUB AKIs are available in all Amazon EC2 regions. There are AKIs for both 32-bit and 64-bit architecture types. Most modern AMIs use a PV-GRUB AKI by default.

We recommend that you always use the latest version of the PV-GRUB AKI, as not all versions of the PV-GRUB AKI are compatible with all instance types. Use the following command to get a list of the PV-GRUB AKIs for the current region:

```
$ ec2-describe-images -o amazon --filter "name=pv-grub-*.gz"
```

Note that PV-GRUB is the only AKI available in the `ap-southeast-2` region. You should verify that any AMI you want to copy to this region is using a version of PV-GRUB that is available in this region.

The following are the current AKI IDs for each region. Register new AMIs using an hd0 AKI.

Note
We continue to provide hd00 AKIs for backward compatibility in regions where they were previously available.

ap-northeast-1, Asia Pacific (Tokyo) region

Image ID	Image Name
`aki-136bf512`	`pv-grub-hd0_1.04-i386.gz`
`aki-176bf516`	`pv-grub-hd0_1.04-x86_64.gz`

ap-southeast-1, Asia Pacific (Singapore) region

Image ID	Image Name
`aki-ae3973fc`	`pv-grub-hd0_1.04-i386.gz`
`aki-503e7402`	`pv-grub-hd0_1.04-x86_64.gz`

ap-southeast-2, Asia Pacific (Sydney) region

Image ID	Image Name
`aki-cd62fff7`	`pv-grub-hd0_1.04-i386.gz`

Image ID	Image Name
aki-c362fff9	pv-grub-hd0_1.04-x86_64.gz

eu-central-1, EU (Frankfurt) region

Image ID	Image Name
aki-3e4c7a23	pv-grub-hd0_1.04-i386.gz
aki-184c7a05	pv-grub-hd0_1.04-x86_64.gz

eu-west-1, EU (Ireland) region

Image ID	Image Name
aki-68a3451f	pv-grub-hd0_1.04-i386.gz
aki-52a34525	pv-grub-hd0_1.04-x86_64.gz

sa-east-1, South America (Sao Paulo) region

Image ID	Image Name
aki-5b53f446	pv-grub-hd0_1.04-i386.gz
aki-5553f448	pv-grub-hd0_1.04-x86_64.gz

us-east-1, US East (N. Virginia) region

Image ID	Image Name
aki-8f9dcae6	pv-grub-hd0_1.04-i386.gz
aki-919dcaf8	pv-grub-hd0_1.04-x86_64.gz

us-gov-west-1, AWS GovCloud (US)

Image ID	Image Name
aki-1fe98d3c	pv-grub-hd0_1.04-i386.gz
aki-1de98d3e	pv-grub-hd0_1.04-x86_64.gz

us-west-1, US West (N. California) region

Image ID	Image Name
aki-8e0531cb	pv-grub-hd0_1.04-i386.gz
aki-880531cd	pv-grub-hd0_1.04-x86_64.gz

us-west-2, US West (Oregon) region

Image ID	Image Name
`aki-f08f11c0`	`pv-grub-hd0_1.04-i386.gz`
`aki-fc8f11cc`	`pv-grub-hd0_1.04-x86_64.gz`

Updating PV-GRUB

We recommend that you always use the latest version of the PV-GRUB AKI, as not all versions of the PV-GRUB AKI are compatible with all instance types. Also, older versions of PV-GRUB are not available in all regions, so if you copy an AMI that uses an older version to a region that does not support that version, you will be unable to boot instances launched from that AMI until you update the kernel image. Use the following procedures to check your instance's version of PV-GRUB and update it if necessary.

To check your PV-GRUB version

1. Find the kernel ID for your instance.

   ```
   $ ec2-describe-instance-attribute instance_id --kernel --region region
   kernel instance_id aki-fc8f11cc
   ```

   The kernel ID for this instance is `aki-fc8f11cc`.
2. View the version information of that kernel ID.

   ```
   $ ec2-describe-images aki-fc8f11cc --region region
   IMAGE aki-fc8f11cc amazon/pv-grub-hd0_1.04-x86_64.gz ...
   ```

   This kernel image is PV-GRUB 1.04. If your PV-GRUB version is not the newest version (as shown in Amazon PV-GRUB Kernel Image IDs (p. 100)), you should update it using the following procedure.

To update your PV-GRUB version

If your instance is using an older version of PV-GRUB, you should update it to the latest version.

1. Identify the latest PV-GRUB AKI for your region and processor architecture from Amazon PV-GRUB Kernel Image IDs (p. 100).
2. Stop your instance. Your instance must be stopped to modify the kernel image used.

   ```
   $ ec2-stop-instances instance_id --region region
   INSTANCE instance_id stopped stopped
   ```

3. Modify the kernel image used for your instance.

   ```
   $ ec2-modify-instance-attribute --kernel kernel_id --region region instance_id
   ```

4. Restart your instance.

   ```
   $ ec2-start-instances --region region instance_id
   ```

Amazon EC2 Instances

If you're new to Amazon EC2, see the following topics to get started:

Before you launch a production environment, you need to answer the following questions.

Q. What purchasing option best meets my needs?

Amazon EC2 supports On-Demand Instances (the default), Spot Instances, and Reserved Instances. For more information, see Amazon EC2 Pricing.

Q. What instance type best meets my needs?

Amazon EC2 provides different instance types to enable you to choose the CPU, memory, storage, and networking capacity that you need to run your applications. For more information, see Instance Types (p. 103).

Q. Which type of root volume meets my needs?

Each instance is backed by Amazon EBS or backed by instance store. Select an AMI based on which type of root volume you need. For more information, see Storage for the Root Device (p. 55).

Q. Would I benefit from using a virtual private cloud?

If you can launch instances in either EC2-Classic or EC2-VPC, you'll need to decide which platform meets your needs. For more information, see Supported Platforms (p. 471) and Amazon EC2 and Amazon Virtual Private Cloud (VPC) (p. 468).

Instance Types

When you launch an instance, the *instance type* that you specify determines the hardware of the host computer used for your instance. Each instance type offers different compute, memory, and storage capabilities. Select an instance type based on the requirements of the application or software that you plan to run on your instance.

Amazon EC2 provides each instance with a consistent and predictable amount of CPU capacity, regardless of its underlying hardware.

Amazon EC2 dedicates some resources of the host computer, such as CPU, memory, and instance storage, to a particular instance. Amazon EC2 shares other resources of the host computer, such as the network and the disk subsystem, among instances. If each instance on a host computer tries to use as much of one of these shared resources as possible, each receives an equal share of that resource. However, when a resource is under-utilized, an instance can consume a higher share of that resource while it's available.

Each instance type provides higher or lower minimum performance from a shared resource. For example, instance types with high I/O performance have a larger allocation of shared resources. Allocating a larger share of shared resources also reduces the variance of I/O performance. For most applications, moderate I/O performance is more than enough. However, for applications that require greater or more consistent I/O performance, consider an instance type with higher I/O performance.

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

To obtain additional, dedicated capacity for Amazon EBS I/O, you can launch some instance types as Amazon EBS–optimized instances. For more information, see Amazon EBS–Optimized Instances (p. 125).

To optimize your instances for high performance computing (HPC) applications, you can launch some instance types in a placement group. For more information, see Placement Groups (p. 126).

When you launch your instance, it uses one of two types of virtualization: paravirtual (PV) or hardware virtual machine (HVM). The virtualization type is determined by the AMI used to launch the instance; some instance types support both PV and HVM while others support only one or the other. 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 PV network and storage drivers for improved performance. For more information, see Virtualization Types (p. 58).

Available Instance Types

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

There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some instance types. For more information, see How many instances can I run in Amazon EC2?

Current Generation Instances

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

Instance Family	Current Generation Instance Types
General purpose	`t2.micro`|`t2.small`|`t2.medium`|`m3.medium`|`m3.large`|`m3.xlarge`|`m3.2xlarge`
Compute optimized	`c3.large`|`c3.xlarge`|`c3.2xlarge`|`c3.4xlarge`|`c3.8xlarge`
Memory optimized	`r3.large`|`r3.xlarge`|`r3.2xlarge`|`r3.4xlarge`|`r3.8xlarge`
Storage optimized	`i2.xlarge`|`i2.2xlarge`|`i2.4xlarge`|`i2.8xlarge`|`hs1.8xlarge`
GPU instances	`g2.2xlarge`

Previous Generation Instances

Amazon Web Services offers previous generation instances for users who have optimized their applications around these instances and have yet to upgrade. We encourage you to use the latest generation of instances to get the best performance, but we will continue to support these previous generation instances after new instances launch. If you are currently using a previous generation instance and would like to see which one would be a suitable upgrade, see Upgrade Paths.

Instance Family	Previous Generation Instance Types
General purpose	`m1.small` | `m1.medium` | `m1.large` | `m1.xlarge`
Compute optimized	`c1.medium` | `c1.xlarge` | `cc2.8xlarge`
Memory optimized	`m2.xlarge` | `m2.2xlarge` | `m2.4xlarge` | `cr1.8xlarge`
Storage optimized	`hi1.4xlarge`
GPU instances	`cg1.4xlarge`
Micro instances	`t1.micro`

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details.

To determine which instance type best meets your needs, we recommend that you launch an instance and use your own benchmark application. Because you pay by the instance hour, it's convenient and inexpensive to test multiple instance types before making a decision. If your needs change, you can resize your instance later on. For more information, see Resizing Your Instance (p. 128).

T2 Instances

T2 instances are designed to provide moderate baseline performance and the capability to burst to significantly higher performance as required by your workload. They are intended for workloads that don't use the full CPU often or consistently, but occasionally need to burst. T2 instances are well suited for general purpose workloads, such as web servers, developer environments, and small databases. For more information about T2 instance pricing and additional hardware details, see Instance Type Details.

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

Topics

- Hardware Specifications (p. 106)
- CPU Credits (p. 106)
- EC2-VPC-only Support (p. 107)
- HVM AMI Support (p. 108)
- Default T2 Instance Limits (p. 108)
- Monitoring Your CPU Credits (p. 108)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details.

CPU Credits

A CPU Credit provides the performance of a full CPU core for one minute. Traditional Amazon EC2 instance types provide fixed performance, while T2 instances provide a baseline level of CPU performance with the ability to burst above that baseline level. The baseline performance and ability to burst are governed by CPU credits.

What is a CPU credit?

One CPU credit is equal to one vCPU running at 100% utilization for one minute. Other combinations of vCPUs, utilization, and time are also equal one CPU credit, such as one vCPU running at 50% utilization for two minutes, or two vCPUs (on t2.medium instances, for example) running at 25% utilization for two minutes.

How are CPU credits earned?

Each T2 instance starts with a healthy initial CPU credit balance and then continuously (at a millisecond-level resolution) receives a set rate of CPU credits per hour, depending on instance size. The accounting process for whether credits are accumulated or spent also happens at a millisecond-level resolution, so you don't have to worry about overspending CPU credits; a short burst of CPU takes a small fraction of a CPU credit.

When a T2 instance uses fewer CPU resources than its base performance level allows (such as when it is idle), the unused CPU credits (or the difference between what was earned and what was spent) are stored in the credit balance for up to 24 hours, building CPU credits for bursting. When your T2 instance requires more CPU resources than its base performance level allows, it uses credits from the CPU credit balance to burst up to 100% utilization. The more credits your T2 instance has for CPU resources, the more time it can burst beyond its base performance level and the better it will perform when more performance is needed.

The following table lists the initial CPU credit allocation received at launch, the rate at which CPU credits are received, the baseline performance level as a percentage of a full core performance, and the maximum CPU credit balance that an instance can accrue.

Instance type	☁ ⚡ 🖥	CPU credits earned per hour	Base performance (CPU utilization)	Maximum CPU credit balance
t2.micro	3	6	10%	144
t2.small	3	12	20%	288
t2.medium	6	24	40%**	576

* There are limits to how many of your T2 instances will launch or start with the initial credit, which by default is set to 100 launches or starts per 24-hour period by region. If you'd like to increase this limit, you can file a customer support limit increase request by using the Amazon EC2 Instance Request Form.

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

The `t2.micro` and `t2.small` instance types launch with an initial balance of 30 CPU credits, and the `t2.medium` instance type launches with 60 CPU credits. This initial credit balance is designed to provide a good startup experience. The maximum credit balance for an instance is equal to the number of CPU credits received per hour times 24 hours. For example, a `t2.micro` instance earns 6 CPU credits per hour and can accumulate a maximum CPU credit balance of 144 CPU credits.

Do CPU credits expire?

Yes. Unused credits (including the initial credit earned at launch time) expire 24 hours after they are earned, and any expired credits are removed from the CPU credit balance at that time. Additionally, the CPU credit balance for an instance does not persist between instance stops and starts; stopping an instance causes it to lose its credit balance entirely, but when it restarts it will receive its initial credit balance again.

For example, if a `t2.small` instance had a CPU utilization of 5% for the hour, it would have used 3 CPU credits (5% of 60 minutes), but it would have earned 12 CPU credits during the hour, so the difference of 9 CPU credits would be added to the CPU credit balance. Any CPU credits in the balance that reached their 24 hour expiration date during that time (which could be as many as 12 credits if the instance was completely idle 24 hours ago) would also be removed from the balance. If the amount of credits expired is greater than those earned, the credit balance will go down; conversely, if the amount of credits expired is fewer than those earned, the credit balance will go up.

What happens if I use all of my credits?

If your instance uses all of its CPU credit balance, performance remains at the baseline performance level. If your instance is running low on credits, your instance's CPU credit consumption (and therefore CPU performance) is gradually lowered to the base performance level over a 15-minute interval, so you will not experience a sharp performance drop-off when your CPU credits are depleted. If your instance consistently uses all of its CPU credit balance, we recommend a larger T2 size or a fixed performance instance type such as M3 or C3.

EC2-VPC-only Support

T2 instances must be launched into an Amazon Virtual Private Cloud (VPC); they are not supported on the EC2-Classic platform. Amazon VPC enables you to launch AWS resources into a virtual network that you've defined. You cannot change the instance type of an existing EC2-Classic instance to a T2 instance type. For more information on EC2-Classic and EC2-VPC, see Supported Platforms (p. 471).

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

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

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

By launching an instance into a VPC, you can leverage a number of features that are available only on the EC2-VPC platform; such as assigning multiple private IP addresses to your instances, or changing your instances' security groups. For more information about the benefits of using a VPC, see Amazon EC2 and Amazon Virtual Private Cloud (VPC) (p. 468). You can take steps to migrate your resources from EC2-Classic to EC2-VPC. For more information, see Migrating from EC2-Classic to a VPC (p. 473).

HVM AMI Support

T2 instances require HVM AMIs.

Default T2 Instance Limits

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

Monitoring Your CPU Credits

You can see the credit balance for each T2 instance presented in the Amazon EC2 per-instance metrics of the CloudWatch console. T2 instances have two metrics, `CPUCreditUsage` and `CPUCreditBalance`. The `CPUCreditUsage` metric indicates the number of CPU credits used during the measurement period. The `CPUCreditBalance` metric indicates the number of unused CPU credits a T2 instance has earned. This balance is depleted during burst time as CPU credits are spent more quickly than they are earned.

The following table describes the new available CloudWatch metrics; for more information on using these metrics in CloudWatch, see View Amazon EC2 Metrics (p. 358).

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

I2 Instances

I2 instances are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications. They are well suited for the following scenarios:

* NoSQL databases (for example, Cassandra and MongoDB)

* Clustered databases

* Online transaction processing (OLTP) systems

You can cluster I2 instances in a placement group. Placement groups provide low latency and high-bandwidth connectivity between the instances within a single Availability Zone. For more information, see Placement Groups (p. 126).

You can enable enhanced networking capabilities for I2 instances. For more information, see Enabling Enhanced Networking on Linux Instances in a VPC (p. 507).

Topics

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details.

I2 Instance Limitations

You must launch an I2 instance using an HVM AMI. We support the following Linux distributions: Amazon Linux, Ubuntu, Red Hat Enterprise Linux, and SUSE Enterprise Linux. If you'd like to use an HVM AMI for a different Linux distribution, let us know by posting to the Amazon EC2 forum.

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

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

SSD Storage

The primary data storage for I2 instances is SSD-based instance storage. Like all instance storage, these volumes persist only for the life of the instance. When you terminate an instance, the applications and data in its instance store is erased. When you use an Amazon EBS-backed AMI, you can start and stop

your instance. However, when you stop an instance, the data stored in the Amazon EBS volume persists, but data in the instance store volumes doesn't persist. We recommend that you regularly back up or replicate the data you've stored in instance storage.

For more information about instance store volumes, see Amazon EC2 Instance Store (p. 581).

SSD I/O Performance

To ensure the best IOPS performance from your I2 instance, we recommend that you use the most recent version of the Amazon Linux AMI, or another Linux AMI with kernel version 3.8 or later. If you use a Linux AMI with kernel version 3.8 or later and utilize all the SSD-based instance store volumes available to the instance, you can get at least the minimum random IOPS (4,096 byte block size) listed in the following table. Otherwise, you'll get lower IOPS performance than what is shown in the table.

Instance Size	Read IOPS	First Write IOPS
i2.xlarge	35,000	35,000
i2.2xlarge	75,000	75,000
i2.4xlarge	175,000	155,000
i2.8xlarge	365,000	315,000

As you fill the SSD-based instance storage for your instance, the number of write IOPS that you can achieve decreases. This is due to the extra work the SSD controller must do to find available space, rewrite existing data, and erase unused space so that it can be rewritten. This process of garbage collection results in internal write amplification to the SSD, expressed as the ratio of SSD write operations to user write operations. This decrease in performance is even larger if the write operations are not in multiples of 4,096 bytes or not aligned to a 4,096-byte boundary. If you write a smaller amount of bytes or bytes that are not aligned, the SSD controller must read the surrounding data and store the result in a new location. This pattern results in significantly increased write amplification, increased latency, and dramatically reduced I/O performance.

SSD controllers can use several strategies to reduce the impact of write amplification. One such strategy is to reserve space in the SSD instance storage so that the controller can more efficiently manage the space available for write operations. This is called *over-provisioning*. The SSD-based instance store volumes provided to an I2 instance don't have any space reserved for over-provisioning. To reduce write amplification, you should leave 10% of the volume unpartitioned so that the SSD controller can use it for over-provisioning. This decreases the storage that you can use, but increases performance.

For Linux instances, you can also use the TRIM command to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information about using TRIM commands, see the documentation for the operating system for your Linux instance.

HI1 Instances

HI1 instances (`hi1.4xlarge`) can deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications. They are well suited for the following scenarios:

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

You can cluster HI1 instances in a placement group. For more information, see Placement Groups (p. 126).

By default, you can run up to two `hi1.4xlarge` instances. If you need more than two `hi1.4xlarge` instances, you can request more using the Amazon EC2 Instance Request Form.

Topics
- Hardware Specifications (p. 111)
- Disk I/O Performance (p. 111)
- SSD Storage (p. 111)

Hardware Specifications

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

For more information about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details.

Disk I/O Performance

Using Linux paravirtual (PV) AMIs, HI1 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. Using hardware virtual machine (HVM) 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 is approximately 2 GB read per second and 1.1 GB write per second.

SSD Storage

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

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

Instance Store with SSD Storage

The `hi1.4xlarge` instances use an Amazon EBS-backed root device. However, their primary data storage is provided by the SSD volumes in the instance store. Like other instance store volumes, these instance store volumes persist only for the life of the instance. Because the root device of the `hi1.4xlarge` instance is Amazon EBS-backed, you can still start and stop your instance. When you stop an instance, your application persists, but your production data in the instance store does not persist. For more information about instance store volumes, see Amazon EC2 Instance Store (p. 581).

Variable Write Performance

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

Write Amplification

Write amplification refers to an undesirable condition associated with flash memory and SSDs, where the actual amount of physical information written is a multiple of the logical amount intended to be written. Because flash memory must be erased before it can be rewritten, the process to perform these operations results in moving (or rewriting) user data and metadata more than once. This multiplying effect increases the number of writes required over the life of the SSD, which shortens the time that it can reliably operate. The `hi1.4xlarge` instances are designed with a provisioning model intended to minimize write amplification.

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

The TRIM Command

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

TRIM support is not available for HI1 instances. For TRIM support, use I2 instances. For more information, see I2 Instances (p. 109).

HS1 Instances

HS1 instances (`hs1.8xlarge`) provide very high storage density and high sequential read and write performance per instance. They are well suited for the following scenarios:

- Data warehousing

- Hadoop/MapReduce

- Parallel file systems

You can cluster HS1 instances in a placement group. For more information, see Placement Groups (p. 126).

By default, you can run up to two HS1 instances. If you need more than two HS1 instances, you can request more using the Amazon EC2 Instance Request Form.

Topics
- Hardware Specifications (p. 112)
- Storage Information (p. 113)

Hardware Specifications

HS1 instances support both Amazon Elastic Block Store (Amazon EBS)-backed and instance store-backed Amazon Machine Images (AMIs). HS1 instances support both paravirtual (PV) and hardware virtual machine (HVM) AMIs.

HS1 instances do not currently support Amazon EBS optimization, but provide high bandwidth networking and can also be used with Provisioned IOPS (SSD) volumes for improved consistency and performance.

For more information about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details.

Storage Information

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

Instance Store with HS1 Instances

HS1 instances support both instance store and Amazon EBS root device volumes. However, even when using an Amazon EBS-backed instance, primary data storage is provided by the hard disk drives in the instance store. Like other instance store volumes, these instance store volumes persist only for the life of the instance. Therefore, when you stop an instance (when using an Amazon EBS-backed root volume), your application persists, but your production data in the instance store does not persist. For more information about instance store volumes, see Amazon EC2 Instance Store (p. 581).

Disk Initialization

If you plan to run an HS1 instance in a steady state for long periods of time, we recommend that you zero the hard disks first for improved performance. This process can take as long as six hours to complete.

Setting the Memory Limit

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

R3 Instances

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

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

You can cluster R3 instances in a placement group. Placement groups provide low latency and high-bandwidth connectivity between the instances within a single Availability Zone. For more information, see Placement Groups (p. 126).

You can enable enhanced networking capabilities for R3 instances. For more information, see Enabling Enhanced Networking on Linux Instances in a VPC (p. 507).

Topics

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details.

Operating System Support

We support the following Linux operating systems: Amazon Linux, Ubuntu, Red Hat Enterprise Linux, and SUSE Enterprise Linux. If you'd like to use an HVM AMI for a different operating system, let us know by posting to the Amazon EC2 forum.

HVM AMI Support

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

Default R3 Instance Limits

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

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

SSD Storage

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

For more information about instance store volumes, see Amazon EC2 Instance Store (p. 581).

SSD I/O Performance

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

To ensure the best IOPS performance from your R3 instance, we recommend that you use the most recent version of the Amazon Linux AMI, or another Linux AMI with a kernel version of 3.8 or later. If you do not use a Linux AMI with a kernel version of 3.8 or later, you will not achieve the maximum IOPS performance available for this instance type.

TRIM Support

You can use the TRIM command on Linux instances to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information about using TRIM commands, see the documentation for the operating system for your instance.

Linux GPU Instances

If you require high parallel processing capability, you'll benefit from using GPU instances, which provide access to NVIDIA GPUs with up to 1,536 CUDA cores and 4 GB of video memory. You can use GPU instances to accelerate many scientific, engineering, and rendering applications by leveraging the Compute Unified Device Architecture (CUDA) or OpenCL parallel computing frameworks. You can also use them for graphics applications, including game streaming, 3-D application streaming, and other graphics workloads.

GPU instances run as HVM-based instances. Hardware virtual machine (HVM) virtualization uses hardware-assist technology provided by the AWS platform. With HVM virtualization, the guest VM runs as if it were on a native hardware platform, except that it still uses paravirtual (PV) network and storage drivers for improved performance. This enables Amazon EC2 to provide dedicated access to one or more discrete GPUs in each GPU instance.

You can cluster GPU instances into a placement group. Placement groups provide low latency and high-bandwidth connectivity between the instances within a single Availability Zone. For more information, see Placement Groups (p. 126).

Topics

- Hardware Specifications (p. 115)
- GPU Instance Limitations (p. 115)
- AMIs for GPU Instances (p. 116)
- Installing the NVIDIA Driver on Linux (p. 116)

For information about Windows GPU Instances, see Windows GPU Instances in the *Amazon EC2 User Guide for Microsoft Windows Instances*.

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details.

GPU Instance Limitations

GPU instances currently have the following limitations:

- They aren't available in every region.
- They must be launched from HVM AMIs.
- They can't access the GPU unless the NVIDIA drivers are installed.
- We limit the number of instances that you can run. For more information, see How many instances can I run in Amazon EC2? in the Amazon EC2 FAQ. To request an increase in these limits, use the following form: Request to Increase Amazon EC2 Instance Limit.

AMIs for GPU Instances

To help you get started, NVIDIA provides AMIs for GPU instances. These reference AMIs include the NVIDIA driver, which enables full functionality and performance of the NVIDIA GPUs. For a list of AMIs with the NVIDIA driver, see AWS Marketplace (NVIDIA GRID).

You can launch a CG1 instance using any HVM AMI.

You can launch a G2 instance using Linux HVM AMIs with the following operating systems: Amazon Linux, SUSE Enterprise Linux, and Ubuntu. If you encounter the following error when launching a G2 instance, contact Customer Service or reach out through the Amazon EC2 forum.

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

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

Installing the NVIDIA Driver on Linux

A GPU instance must have the appropriate NVIDIA driver. The NVIDIA driver you install must be compiled against the kernel that you intend to run on your instance.

Amazon provides AMIs with updated and compatible builds of the NVIDIA kernel drivers for each official kernel upgrade in the AWS Marketplace. If you decide to use a different NVIDIA driver version than the one that Amazon provides, or decide to use a kernel that's not an official Amazon build, you must uninstall the Amazon-provided NVIDIA packages from your system to avoid conflicts with the versions of the drivers that you are trying to install.

Use this command to uninstall Amazon-provided NVIDIA packages:

```
[ec2-user ~]$ sudo yum erase nvidia cudatoolkit
```

The Amazon-provided CUDA toolkit package has dependencies on the NVIDIA drivers. Uninstalling the NVIDIA packages erases the CUDA toolkit. You must reinstall the CUDA toolkit after installing the NVIDIA driver.

You can download NVIDIA drivers from http://www.nvidia.com/Download/Find.aspx. Select the appropriate driver for your instance:

G2 Instances

Product Type	GRID
Product Series	GRID Series
Product	GRID K520
Operating System	Linux 64-bit
Recommended/Beta	Recommended/Certified

CG1 Instances

Product Type	Tesla
Product Series	M-Class
Product	M2050
Operating System	Linux 64-bit
Recommended/Beta	Recommended/Certified

For more information about installing and configuring the driver, open the **ADDITIONAL INFORMATION** tab on the download page for the driver on the NVIDIA website and click the README link.

Install the NVIDIA Driver Manually

To install the driver for an Amazon Linux AMI

1. Run the **yum update** command to get the latest versions of packages for your instance.

    ```
    [ec2-user ~]$ sudo yum update -y
    ```

2. Reboot your instance to load the latest kernel version.

    ```
    [ec2-user ~]$ sudo reboot
    ```

3. Reconnect to your instance after it has rebooted.
4. Install the "Development tools" package group.

    ```
    [ec2-user ~]$ sudo yum groupinstall -y "Development tools"
    ```

5. Make sure the `kernel-devel` package is installed and matches the version of the kernel you are currently running.

    ```
    [ec2-user ~]$ sudo yum install kernel-devel-`uname -r`
    ```

6. Download the driver package that you identified earlier. For example, the following command downloads the 340.46 version of the G2 instance driver.

    ```
    [ec2-user ~]$ wget http://us.download.nvidia.com/XFree86/Linux-
    x86_64/340.46/NVIDIA-Linux-x86_64-340.46.run
    ```

7. Run the self-install script to install the NVIDIA driver. For example:

    ```
    [ec2-user ~]$ sudo /bin/bash ./NVIDIA-Linux-x86_64-340.46.run
    ```

8. Reboot the instance. For more information, see Reboot Your Instance (p. 312).
9. Confirm that the driver is functional. The response for the following command lists the installed NVIDIA driver version and details about the GPUs.

```
[ec2-user ~]$ nvidia-smi -q | head

==============NVSMI LOG==============

Timestamp                           : Thu Oct  2 17:28:29 2014
Driver Version                      : 340.46

Attached GPUs                       : 1
GPU 0000:00:03.0
    Product Name                    : GRID K520
    Product Brand                   : Grid
```

T1 Micro Instances

T1 Micro instances (`t1.micro`) provide a small amount of consistent CPU resources and allow you to increase CPU capacity in short bursts when additional cycles are available. They are well suited for lower throughput applications and websites that require additional compute cycles periodically.

> **Note**
> The `t1.micro` is a previous generation instance and it has been replaced by the `t2.micro`, which has a much better performance profile. We recommend using the `t2.micro` instance type instead of the `t1.micro`. For more information, see T2 Instances (p. 105).

The `t1.micro` instance is available as an Amazon EBS-backed instance only.

This documentation describes how `t1.micro` instances work so that you can understand how to apply them. It's not our intent to specify exact behavior, but to give you visibility into the instance's behavior so you can understand its performance.

Topics
- Hardware Specifications (p. 118)
- Optimal Application of T1 Micro Instances (p. 118)
- Available CPU Resources During Spikes (p. 120)
- When the Instance Uses Its Allotted Resources (p. 121)
- Comparison with the m1.small Instance Type (p. 122)
- AMI Optimization for Micro Instances (p. 124)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details.

Optimal Application of T1 Micro Instances

A `t1.micro` instance provides spiky CPU resources for workloads that have a CPU usage profile similar to what is shown in the following figure.
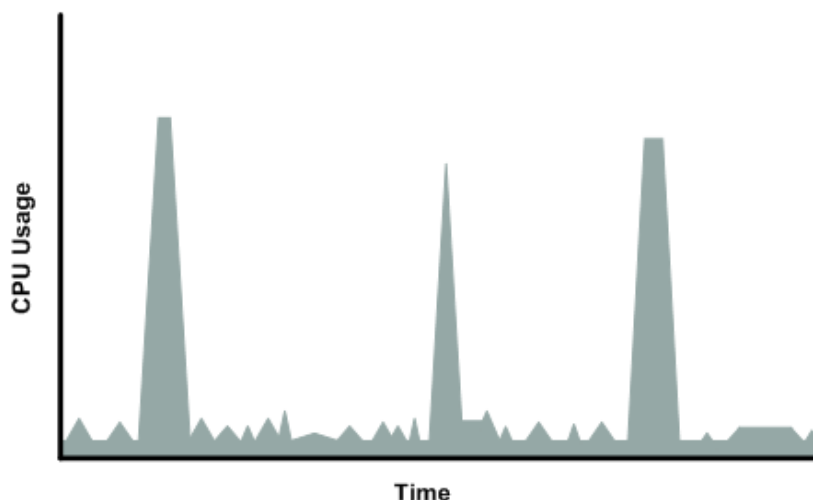
The instance is designed to operate with its CPU usage at essentially only two levels: the normal low background level, and then at brief spiked levels much higher than the background level. We allow the instance to operate at up to 2 EC2 compute units (ECUs) (one ECU provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor). The ratio between the maximum level and the background level is designed to be large. We designed t1.micro instances to support tens of requests per minute on your application. However, actual performance can vary significantly depending on the amount of CPU resources required for each request on your application.

Your application might have a different CPU usage profile than that described in the preceding section. The next figure shows the profile for an application that isn't appropriate for a t1.micro instance. The application requires continuous data-crunching CPU resources for each request, resulting in plateaus of CPU usage that the t1.micro instance isn't designed to handle.
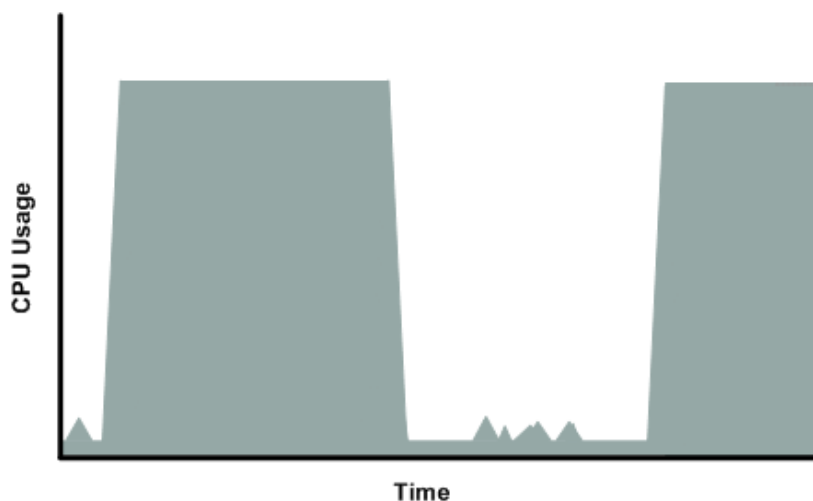
The next figure shows another profile that isn't appropriate for a t1.micro instance. Here the spikes in CPU use are brief, but they occur too frequently to be serviced by a micro instance.
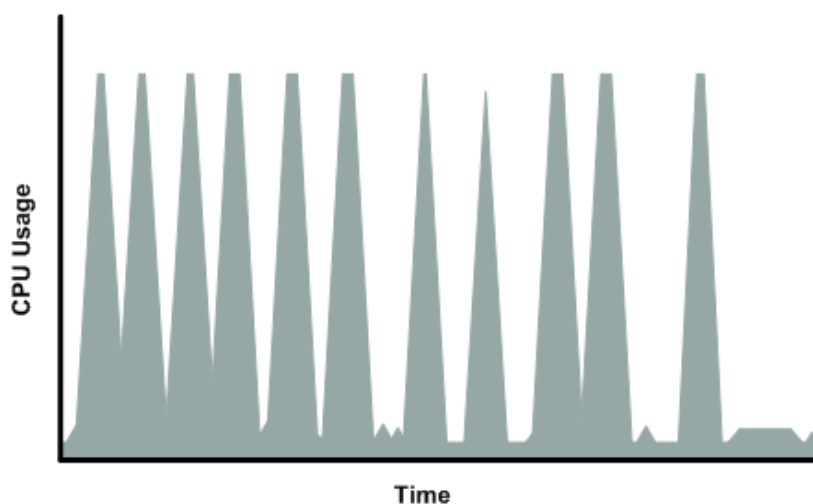
The next figure shows another profile that isn't appropriate for a `t1.micro` instance. Here the spikes aren't too frequent, but the background level between spikes is too high to be serviced by a `t1.micro` instance.
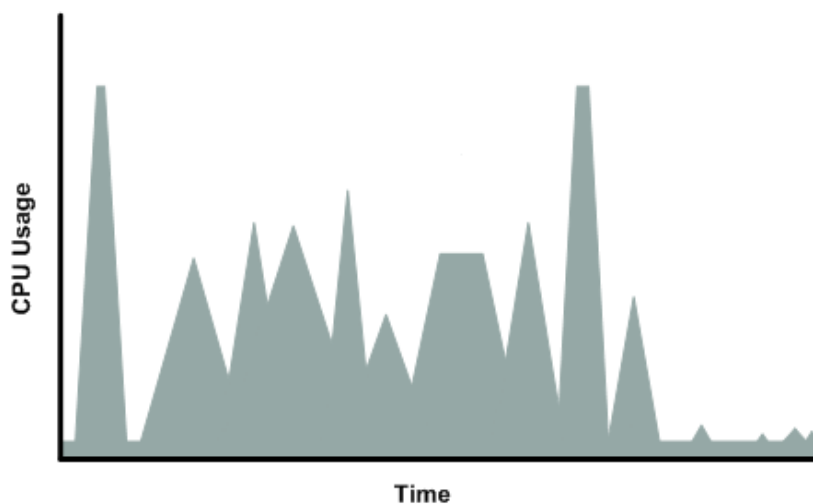
In each of the preceding cases of workloads not appropriate for a `t1.micro` instance, we recommend that you consider using a different instance type. For more information about instance types, see Instance Types (p. 103).

Available CPU Resources During Spikes

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

When the Instance Uses Its Allotted Resources

We expect your application to consume only a certain amount of CPU resources in a period of time. If the application consumes more than your instance's allotted CPU resources, we temporarily limit the instance so it operates at a low CPU level. If your instance continues to use all of its allotted resources, its performance will degrade. We will increase the time that we limit its CPU level, thus increasing the time before the instance is allowed to burst again.

If you enable CloudWatch monitoring for your `t1.micro` instance, you can use the "Avg CPU Utilization" graph in the AWS Management Console to determine whether your instance is regularly using all its allotted CPU resources. We recommend that you look at the maximum value reached during each given period. If the maximum value is 100%, we recommend that you use Auto Scaling to scale out (with additional `t1.micro` instances and a load balancer), or move to a larger instance type. For more information, see the Auto Scaling Developer Guide.

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

The next figure shows the situation with the long plateaus of data-crunching CPU usage. The CPU hits the maximum allowed level and stays there until the instance's allotted resources are consumed for the period. At that point, we limit the instance to operate at the low background level, and it operates there until we allow it to burst above that level again. The instance again stays there until the allotted resources are consumed and we limit it again (not seen on the graph).
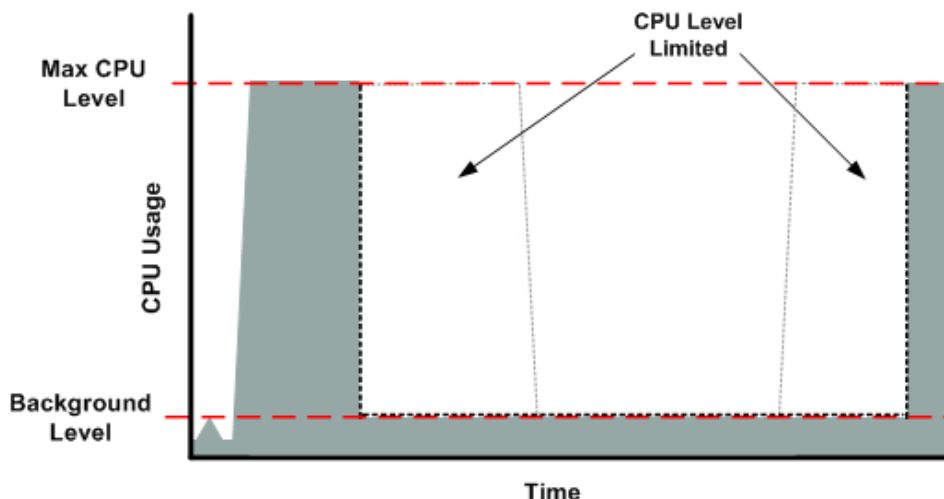
The next figure shows the situation where the requests are too frequent. The instance uses its allotted resources after only a few requests and so we limit it. After we lift the restriction, the instance maxes out its CPU usage trying to keep up with the requests, and we limit it again.
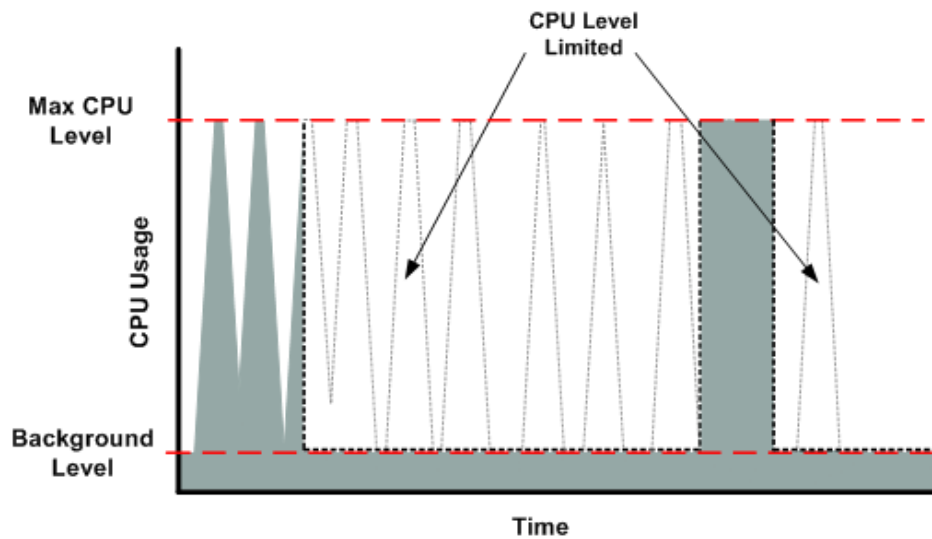
The next figure shows the situation where the background level is too high. Notice that the instance doesn't have to be operating at the maximum CPU level for us to limit it. We limit the instance when it's operating above the normal background level and has consumed its allotted resources for the given period. In this case (as in the preceding one), the instance can't keep up with the work, and we limit it again.
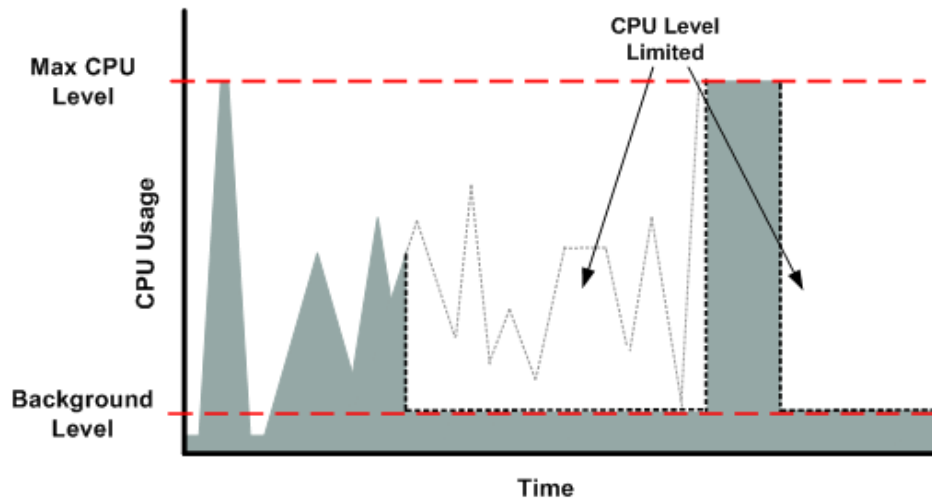
Comparison with the m1.small Instance Type

The `t1.micro` instance provides different levels of CPU resources at different times (up to 2 ECUs). By comparison, the `m1.small` instance type provides 1 ECU at all times. The following figure illustrates the difference.
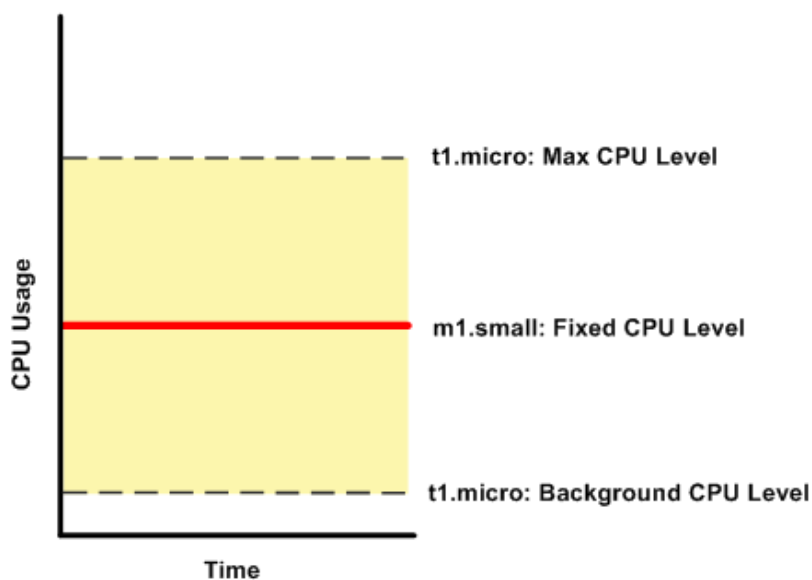
The following figures compare the CPU usage of a `t1.micro` instance with an `m1.small` instance for the various scenarios we've discussed in the preceding sections.

The first figure that follows shows an optimal scenario for a `t1.micro` instance (the left graph) and how it might look for an `m1.small` instance (the right graph). In this case, we don't need to limit the `t1.micro` instance. The processing time on the `m1.small` instance would be longer for each spike in CPU demand compared to the `t1.micro` instance.
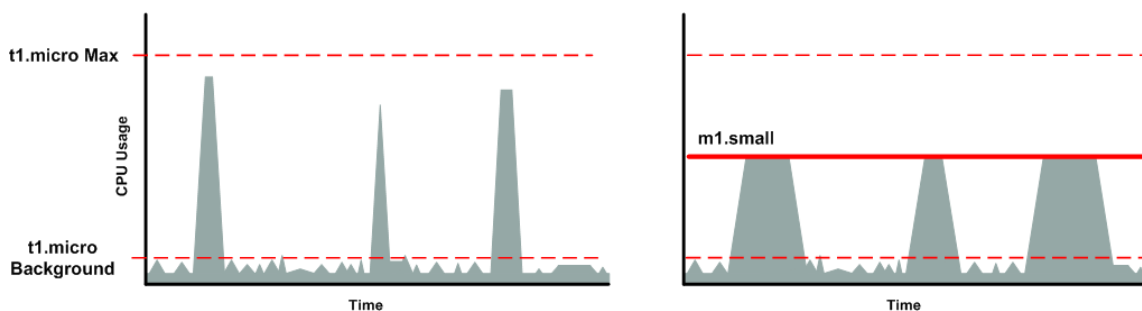
The next figure shows the scenario with the data-crunching requests that used up the allotted resources on the `t1.micro` instance, and how they might look with the `m1.small` instance.
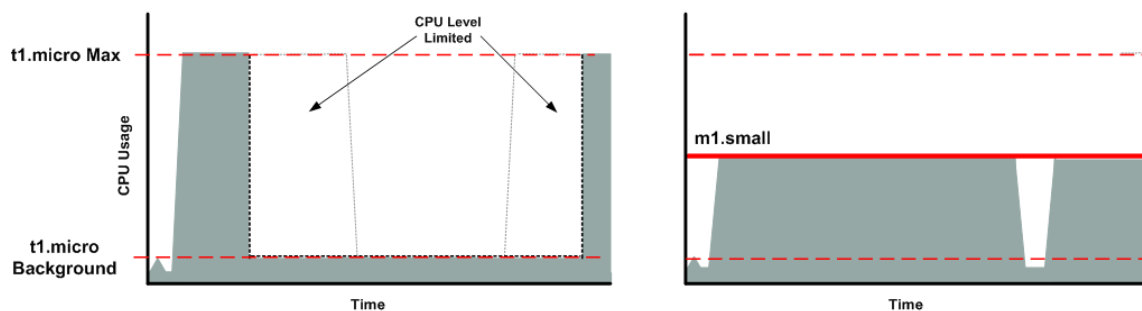
The next figure shows the frequent requests that used up the allotted resources on the `t1.micro` instance, and how they might look on the `m1.small` instance.
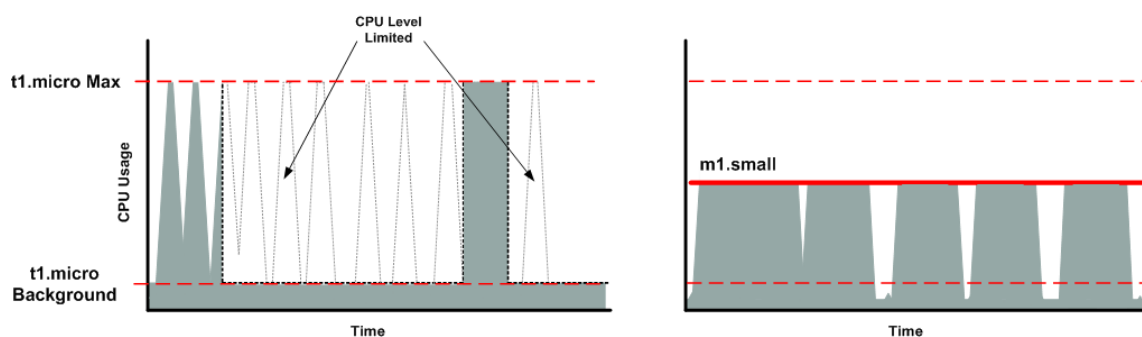
The next figure shows the situation where the background level used up the allotted resources on the `t1.micro` instance, and how it might look on the `m1.small` instance.
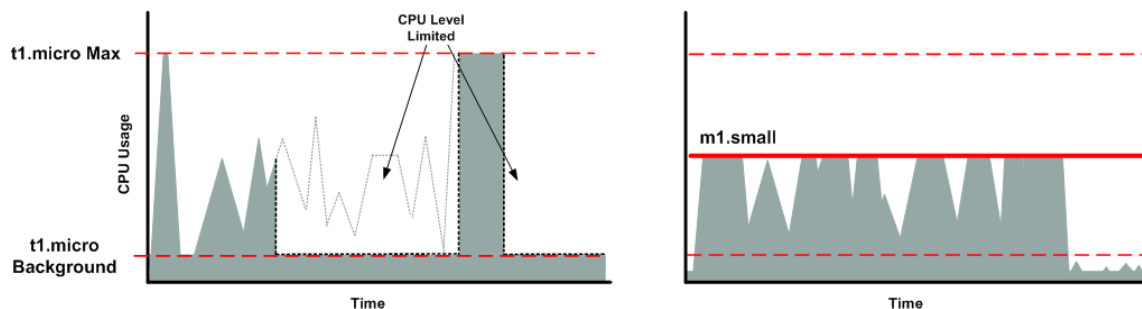
AMI Optimization for Micro Instances

We recommend that you follow these best practices when optimizing an AMI for the `t1.micro` instance type:

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

You can optimize performance using swap space and virtual memory (for example, by setting up swap space in a separate partition from the root file system).

Amazon EBS–Optimized Instances

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

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

Amazon EBS–optimized instances deliver dedicated throughput to Amazon EBS, with options between 500 Mbps and 2,000 Mbps, depending on the instance type you use. When attached to an Amazon EBS–optimized instance, General Purpose (SSD) volumes are designed to deliver within 10 percent of their of the baseline and burst performance 99.9 percent of the time in a given year, and Provisioned IOPS (SSD) volumes are designed to deliver within 10 percent of their provisioned performance 99.9 percent of the time in a given year. For more information, see Amazon EBS Volume Types (p. 523).

The following table shows which instance types are available to be launched as EBS-optimized, the dedicated throughput to Amazon EBS, the maximum amount of IOPS the instance can support if you are using a 16 KB I/O size, and the approximate maximum bandwidth available on that connection in MB/s. Be sure to choose an EBS-optimized instance that provides more dedicated EBS throughput than your application needs; otherwise, the EBS to EC2 connection will become a performance bottleneck.

Instance Type	Dedicated EBS Throughput (Mbps)*	Max 16K IOPS**	Max Bandwidth (MB/s)**
c1.xlarge	1,000	8,000	125
c3.xlarge	500	4,000	62.5
c3.2xlarge	1,000	8,000	125
c3.4xlarge	2,000	16,000	250
g2.2xlarge	1,000	8,000	125
i2.xlarge	500	4,000	62.5
i2.2xlarge	1,000	8,000	125
i2.4xlarge	2,000	16,000	250
m1.large	500	4,000	62.5
m1.xlarge	1,000	8,000	125
m2.2xlarge	500	4,000	62.5
m2.4xlarge	1,000	8,000	125
m3.xlarge	500	4,000	62.5
m3.2xlarge	1,000	8,000	125
r3.xlarge	500	4,000	62.5
r3.2xlarge	1,000	8,000	125
r3.4xlarge	2,000	16,000	250

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

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

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

Placement Groups

A *placement group* is a logical grouping of instances within a single Availability Zone. Using placement groups enables applications to participate in a low-latency, 10 Gbps network. Placement groups are recommended for applications that benefit from low network latency, high network throughput, or both. To provide the lowest latency, and the highest packet-per-second network performance for your placement group, choose an instance type that supports enhanced networking. For more information, see Enhanced Networking (p. 507).

First, you create a placement group and then you launch multiple instances into the placement group. We recommend that you launch the number of instances that you need in the placement group in a single launch request and that you use the same instance type for all instances in the placement group. If you try to add more instances to the placement group later, or if you try to launch more than one instance type in the placement group, you increase your chances of getting an insufficient capacity error.

If you stop an instance in a placement group and then start it again, it still runs in the placement group. However, the start fails if there isn't enough capacity for the instance.

If you receive a capacity error when launching an instance in a placement group, stop and restart the instances in the placement group, and then try the launch again.

Topics

- Placement Group Limitations (p. 126)
- Launching Instances into a Placement Group (p. 127)
- Deleting a Placement Group (p. 128)

Placement Group Limitations

Placement groups have the following limitations:

- A placement group can't span multiple Availability Zones.
- The name you specify for a placement group a name must be unique within your AWS account.
- The following are the only instance types that you can use when you launch an instance into a placement group:
  - Compute optimized: `c3.large` | `c3.xlarge` | `c3.2xlarge` | `c3.4xlarge` | `c3.8xlarge` | `cc2.8xlarge`
  - GPU: `cg1.4xlarge` | `g2.2xlarge`
  - Memory optimized: `cr1.8xlarge` | `r3.large` | `r3.xlarge` | `r3.2xlarge` | `r3.4xlarge` | `r3.8xlarge`
  - Storage optimized: `hi1.4xlarge` | `hs1.8xlarge` | `i2.xlarge` | `i2.2xlarge` | `i2.4xlarge` | `i2.8xlarge`
- Although launching multiple instance types into a placement group is possible, this reduces the likelihood that the required capacity will be available for your launch to succeed. We recommend using the same instance type for all instances in a placement group.

- You can't merge placement groups. Instead, you must terminate the instances in one placement group, and then relaunch those instances into the other placement group.
- A placement group can span peered VPCs; however, you will not get full-bisection bandwidth between instances in peered VPCs. For more information about VPC peering connections, see VPC Peering in the *Amazon VPC User Guide*.
- You can't move an existing instance into a placement group. You can create an AMI from your existing instance, and then launch a new instance from the AMI into a placement group.

Launching Instances into a Placement Group

We suggest that you create an AMI specifically for the instances that you'll launch into a placement group.

To launch an instance into a placement group using the console

1. Open the Amazon EC2 console.
2. Create an AMI for your instances.

    a. From the Amazon EC2 dashboard, click **Launch Instance**. After you complete the wizard, click **Launch**.

    b. Connect to your instance. (For more information, see Connect to Your Linux Instance (p. 299).)

    c. Install software and applications on the instance, copy data, or attach additional Amazon EBS volumes.

    d. (Optional) If your instance type supports enhanced networking, ensure that this feature is enabled by following the procedures in Enabling Enhanced Networking on Linux Instances in a VPC (p. 507).

    e. In the navigation pane, click **Instances**, select your instance, click **Actions**, and then click **Create Image**. Provide the information requested by the **Create Image** dialog box, and then click **Create Image**.

    f. (Optional) You can terminate this instance if you have no further use for it.

3. Create a placement group.

    a. In the navigation pane, click **Placement Groups**.

    b. Click **Create Placement Group**.

    c. In the **Create Placement Group** dialog box, provide a name for the placement group that is unique in the AWS account you're using, and then click **Create**.

        When the status of the placement group is `available`, you can launch instances into the placement group.

4. Launch your instances into the placement group.

    a. In the navigation pane, click **Instances**.

    b. Click **Launch Instance**. Complete the wizard as directed, taking care to select the following:

        - The AMI that you created
        - The number of instances that you'll need
        - The placement group that you created

To create a placement group using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- create-placement-group (AWS CLI)
- ec2-create-placement-group (Amazon EC2 CLI)
- New-EC2PlacementGroup (AWS Tools for Windows PowerShell)

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

Deleting a Placement Group

You can delete a placement group if you need to replace it or no longer need a placement group. Before you can delete your placement group, you must terminate all instances that you launched into the placement group.

To delete a placement group using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**.
3. Select and terminate all instances in the placement group. (You can verify that the instance is in a placement group before you terminate it by checking the value of **Placement Group** in the details pane.)
4. In the navigation pane, click **Placement Groups**.
5. Select the placement group, and then click **Delete Placement Group**.
6. When prompted for confirmation, click **Yes, Delete**.

To delete a placement group using the command line

You can use one of the following sets of commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- terminate-instances and delete-placement-group (AWS CLI)
- ec2-terminate-instances and ec2-delete-placement-group (Amazon EC2 CLI)
- Stop-EC2Instance and Remove-EC2PlacementGroup(AWS Tools for Windows PowerShell)

Resizing Your Instance

As your needs change, you might find that your instance is over-utilized (the instance type is too small) or under-utilized (the instance type is too large). If this is the case, you can change the size of your instance. For example, if your `t1.micro` instance is too small for its workload, you can change it to an `m1.small` instance.

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

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

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

For more information about root device volumes, see Storage for the Root Device (p. 55).

Topics
- Limitations for Resizing Instances (p. 129)
- Resizing an Amazon EBS–backed Instance (p. 129)
- Resizing an Instance Store-backed Instance (p. 130)

Limitations for Resizing Instances

Amazon Machine Images use one of two types of virtualization: paravirtual (PV) or hardware virtual machine (HVM). All current generation instance types support HVM AMIs. Some previous generation instance types, such as T1, C1, M1, and M2 do not support Linux HVM AMIs. Some current generation instance types, such as T2, I2, R3, G2, CR1, and CC2 do not support PV AMIs. You cannot change the virtualization type of an instance or an AMI; an instance can only be resized to an instance type that supports its method of virtualization, and AMIs can only be launched on instance types that support their method of virtualization. For more information, see Virtualization Types (p. 58).

T2 instances must be launched into a VPC using HVM AMIs; they are not supported on the EC2-Classic platform and they do not support PV AMIs. If your account supports EC2-Classic and you have not created any VPCs, you cannot change your instance type to T2 in the console. If your instance uses HVM virtualization and it was launched into a VPC, then you can resize that instance to a T2 instance. For more information, see T2 Instances (p. 105), Amazon EC2 and Amazon Virtual Private Cloud (VPC) (p. 468), and Virtualization Types (p. 58).

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

You cannot add instance store volumes when you resize your instance; instance store volumes may only be added at launch time. If you want to add instance store volumes, consider creating an AMI from your instance and launching a new instance from that AMI with instance store volumes. For more information, see Amazon EC2 Instance Store (p. 581).

Resizing an Amazon EBS–backed Instance

You must stop your Amazon EBS–backed instance before you can change its instance type. When you stop and start an instance, we move it to new hardware. If the instance is running in EC2-Classic, we give it new public and private IP addresses, and disassociate any Elastic IP address that's associated with the instance. Therefore, to ensure that your users can continue to use the applications that you're hosting on your instance uninterrupted, you must re-associate any Elastic IP address after you restart your instance. For more information, see Stop and Start Your Instance (p. 309).

Use the following procedure to resize an Amazon EBS–backed instance using the AWS Management Console.

To resize an Amazon EBS–backed instance

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**, and select the instance.
3. [EC2-Classic] If the instance has an associated Elastic IP address, write down the Elastic IP address and the instance ID shown in the details pane.
4. Click **Actions**, and then click **Stop**.

5. In the confirmation dialog box, click **Yes, Stop**. It can take a few minutes for the instance to stop.

    [EC2-Classic] When the instance state becomes `stopped`, the **Elastic IP**, **Public DNS**, **Private DNS**, and **Private IPs** fields in the details pane are blank to indicate that the old values are no longer associated with the instance.

6. With the instance still selected, click **Actions**, and then click **Change Instance Type**. Note that this action is disabled if the instance state is not `stopped`.

7. In the **Change Instance Type** dialog box, in the **Instance Type** list, select the type of instance that you need, and then click **Apply**.

8. To restart the stopped instance, select the instance, click **Actions**, and then click **Start**.

9. In the confirmation dialog box, click **Yes, Start**. It can take a few minutes for the instance to enter the `running` state.

    [EC2-Classic] When the instance state is `running`, the **Public DNS**, **Private DNS**, and **Private IPs** fields in the details pane contain the new values that we assigned to the instance.

10. [EC2-Classic] If your instance had an associated Elastic IP address, you must reassociate it as follows:

    a. In the navigation pane, click **Elastic IPs**.
    b. Select the Elastic IP address that you wrote down before you stopped the instance.
    c. Click **Associate Address**.
    d. Select the instance ID that you wrote down before you stopped the instance, and then click **Associate**.

Resizing an Instance Store-backed Instance

You can create an image from your current instance, launch a new instance from this image with the instance type you need, and then terminate the original instance that you no longer need. To ensure that your users can continue to use the applications that you're hosting on your instance uninterrupted, you must take any Elastic IP address that you've associated with your current instance and associate it with the new instance.

To resize an instance store-backed instance

1. (Optional) If the instance you are resizing has an associated Elastic IP address, record the Elastic IP address now so that you can associate it with the resized instance later.

2. Create an AMI from your instance store-backed instance by satisfying the prerequisites and following the procedures in Creating an Instance Store-Backed Linux AMI (p. 77). When you are finished creating a new AMI from your instance, return to this procedure.

3. Open the Amazon EC2 console and in the navigation pane, select **AMIs**. From the filter lists, select **Owned by me**, and select the image you created in the previous step. Notice that **AMI Name** is the name that you specified when you registered the image and **Source** is your Amazon S3 bucket.

    > **Note**
    > If you do not see the AMI that you created in the previous step, make sure that the console displays the region that you created your AMI in.

4. Click **Launch**. When you specify options in the launch wizard, be sure to specify the new instance type that you need. It can take a few minutes for the instance to enter the `running` state.

5. (Optional) If the instance that you started with had an associated Elastic IP address, you must associate it with the new instance as follows:

    a. In the navigation pane, click **Elastic IPs**.
    b. Select the Elastic IP address that you recorded at the beginning of this procedure.

        c.   Click **Associate Address**.

        d.   Select the instance ID of the new instance, and then click **Associate**.

6.   (Optional) You can terminate the instance that you started with, if it's no longer needed. Select the instance and check its instance ID against the instance ID that you wrote down at the beginning of this procedure to verify that you are terminating the correct instance. Click **Actions**, and then click **Terminate**.

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 based on the last fulfilled bid price, 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.

> **Note**
> You can run Amazon EC2 Spot Instances on the following platforms: Amazon Linux/UNIX, SUSE Linux Enterprise Server, and Windows (without SQL Server). Spot Instances are not available in the AWS Free Usage Tier program.

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.

For product and pricing information, see the following pages:

- Amazon EC2 Spot Instances
- AWS Service Pricing Overview
- Amazon EC2 On-Demand Instances Pricing

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.

What's New in Spot Instances

Here's a quick look at what's new in Spot Instances:

Getting Started with Spot Instances

In this section, you will get started using Spot Instances. You'll walk through what you need to know before you begin and the prerequisites you need to get started. You'll learn about the tools you will use for specific Spot Instance tasks, and then you'll step through the main tasks.

If you are new to Spot Instances, take a look at Prerequisites for Using Spot Instances (p. 134) 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 items in the following list to get going.

Topics

Before requesting a Spot Instance, consider configuring your Amazon Machine Image (AMI) so that your application does the following tasks:

- 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 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 AMI (p. 54).

Prerequisites for Using Spot Instances

To work with Amazon EC2 Spot Instances, read and complete the instructions described in Getting Started with Amazon EC2 Linux Instances (p. 26), which provides information on creating your Amazon EC2 account and credentials.

In addition, there are a variety of tools you can use to work with Spot Instances. Whichever you choose—the AWS Management Console, the AWS Command Line Interface (AWS CLI), the Amazon EC2 command line interface (EC2 CLI), or the Amazon EC2 application programming interface (EC2 API)—you will have specific tools for assessing Spot price history, submitting Spot Instance requests (also called *bids*), and managing 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.

In this topic, you get an overview of the AWS Management Console, and the AWS CLI and Amazon EC2 CLI tools available for Spot Instances. We assume that you have read and completed the instructions described in the following guides. They walk you through setting up your environment for use with the tools:
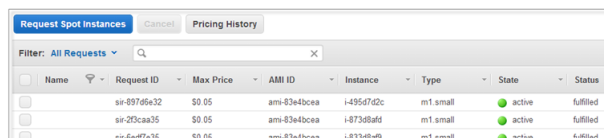
- AWS Management Console—Getting Started with AWS Management Console
- AWS CLI—AWS Command Line Interface User Guide
- Amazon EC2 CLI—Setting Up the Amazon EC2 Tools

For information about using the Amazon EC2 API, see Making API Requests (p. 665). For information about API actions, see the Amazon EC2 API Reference.

Using the AWS Management Console for Spot Instances

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 **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.
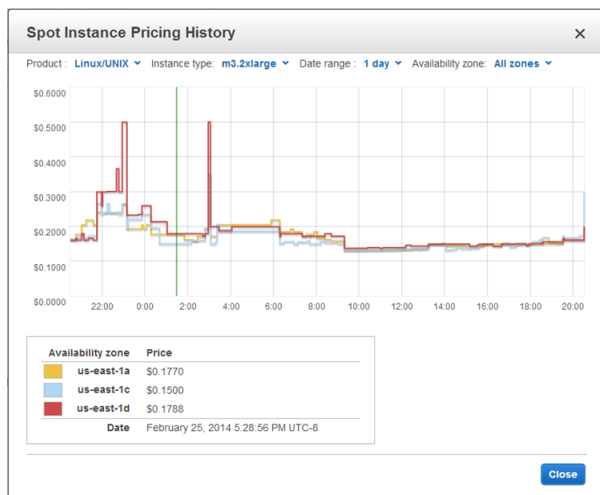
- Use the **Request Spot Instances** page to submit a Spot Instance request and specify the details of the instance to be launched when your request succeeds.

- Use the **Instances** page to manage the instances launched when your Spot request succeeds.

Using the AWS CLI and the EC2 CLI Tools for Spot Instances

The AWS CLI and the Amazon EC2 CLI tools have commands that are specifically designed for managing Spot requests. The following table lists the commands you use for Spot request tasks. To manage the instances launched when your Spot request is fulfilled, use the same commands that you use for On-Demand EC2 instances.

Task	AWS CLI	Amazon EC2 CLI
Viewing Spot Instance Pricing History (p. 136).	`aws ec2 describe-spot-price-history`	`ec2-describe-spot-price-history`

Task	AWS CLI	Amazon EC2 CLI
Finding Running Spot Instances (p. 141).	`aws ec2 describe-spot-instance-requests`	ec2-describe-spot-instance-requests
Creating a Spot Instance Request (p. 138).	`aws ec2 request-spot-instances`	ec2-request-spot-instances
Subscribe to Your Spot Instance Data Feed (p. 181).	`aws ec2 create-spot-datafeed-subscription`	ec2-create-spot-datafeed-subscription
Data Feed Filename and Format (p. 180).	`aws ec2 describe-spot-datafeed-subscription`	ec2-describe-spot-datafeed-subscription
Delete a Spot Instance Data Feed (p. 182).	`aws ec2 delete-spot-datafeed-subscription`	ec2-delete-spot-datafeed-subscription
Canceling Spot Instance Requests (p. 143).	`aws ec2 cancel-spot-instance-requests`	ec2-cancel-spot-instance-requests

Using AWS Java SDK for Spot Instances

Java developers can go to the *AWS SDK for Java* to consult the Java tutorials on Spot Instances:

- Tutorial: Amazon EC2 Spot Instances
- Tutorial: Advanced Amazon EC2 Spot Request Management

Viewing Spot Instance Pricing 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 Instance Pricing 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. 134).
> 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 **Pricing History**.

   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 **Availability Zone** 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.
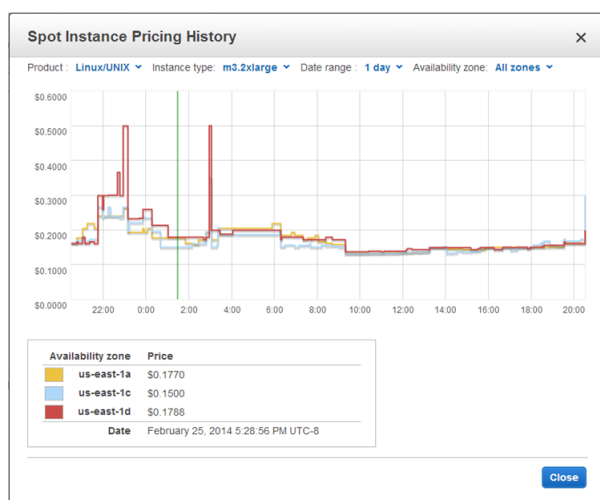
4. To view Spot pricing details, move your cursor across the graph to specific points in time. A vertical line follows the movement of your cursor, and the table below the graph displays the Spot prices for different Availability Zones and the date and time pointed to by your cursor and the vertical line.

5. 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:

```
$ ec2-describe-spot-price-history -H --instance-type m1.xlarge
```

Amazon EC2 returns output similar to the following:

```
SPOTINSTANCEPRICE     0.384000     2011-05-25T11:37:48-0800     m1.xlarge
Windows us-east-1a
SPOTINSTANCEPRICE     0.384000     2011-05-25T11:37:48-0800     m1.xlarge
Windows us-east-1d
…
```

```
SPOTINSTANCEPRICE    0.242000    2011-04-18T14:39:14-0800    m1.xlarge    SUSE
 Linux  us-east-1d
SPOTINSTANCEPRICE    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 EC2 Command Line Reference*.

Viewing Spot Price History Using the API

To view the Spot Price history using the API, use the DescribeSpotPriceHistory action.

What do you want to do next?

- Creating a Spot Instance Request (p. 138)
- Finding Running Spot Instances (p. 141)
- Launching Spot Instances in Amazon Virtual Private Cloud (p. 177)

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. 150).

In this section, we discuss how to create requests for Amazon EC2 Spot Instances using the AWS Management Console or the CLI. To create a Spot Instance request using the API, use the RequestSpotInstances action. For more information about requesting Spot Instances, see the following:

- Amazon EC2 Spot Instance Limits (p. 146)
- Types of Spot Instance Requests (p. 149)
- Launching Spot Instances in Launch Groups and Availability Zone Groups (p. 149)
- Spot Instance Request States (p. 148)
- Launching Spot Instances in Amazon Virtual Private Cloud (p. 177)

> **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. 134).

AWS Management Console

To create a Spot Instance request

1. From the Amazon EC2 console, click **Spot Requests** in the navigation pane.

2. Click **Request Spot Instances**.

   Except for the Spot Instance options that you specify to configure your Spot Instances, the process
   for requesting Spot Instances is the same as the process for launching On-Demand instances. You
   go through the same steps when you click **Launch Instance** from the **Instances** page: Choose an
   Amazon Machine Image (AMI) and an instance type, and configure the details of the instance. For
   information, see Launch Your Instance (p. 289).

3. On the **Configure Instance Details** screen, **Request Spot Instances** is checked as the **Purchasing
   Option** by default if you get to this configure screen from the **Spot Requests** page. It is only when
   the **Request Spot Instances** option is selected that the options related to Spot Instance requests
   are displayed.

   Specify the **Maximum price** you are willing to pay for your Spot Instance. This is a required option.

   Notice that the current Spot Price for each of the Availability Zones in your region is listed. Use this
   information as a guide on the price to specify so that your request is fulfilled. Your instance launches
   and runs if your bid price exceeds the Spot Price.

   > **Note**
   > You cannot change the parameters of your Spot request, including the Spot price, after
   > you've submitted the request.

   To explore Spot Price trends, click **Pricing History** in the **Spot Requests** page. For information,
   see Viewing Spot Instance Pricing History (p. 136).

4. There are several details that you can define optionally. Select the options you want to define your
   Spot request and to configure your Spot Instance settings.

Option	Description
Request valid from and **Request valid to**	Defines the validity period of a Spot request. The validity period, beginning with **Request valid from** through **Request valid to**, 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.

Option	Description
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 Zone Groups (p. 149).
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 Zone Groups (p. 149).

5. Continue with the wizard as you normally would when launching instances. For information, see Launch Your Instance (p. 289).

    If your request specified multiple instances, Amazon EC2 creates a separate Spot Instance request for each instance. View these requests on the **Spot Requests** page.

Amazon EC2 Command Line Interface (CLI) Tools

To create a Spot Price request

*   Enter the following command:

```
$ 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:

```
$ 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).

Requesting Spot Instances Using the API

To request Spot Instances using the API, use the RequestSpotInstances action.

What do you want to do next?

- Finding Running Spot Instances (p. 141)
- Canceling Spot Instance Requests (p. 143)
- Planning for Interruptions (p. 161)
- Launching Spot Instances in Amazon Virtual Private Cloud (p. 177)
- Tagging Spot Instance Requests (p. 156)
- 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. 134).

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.

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.)
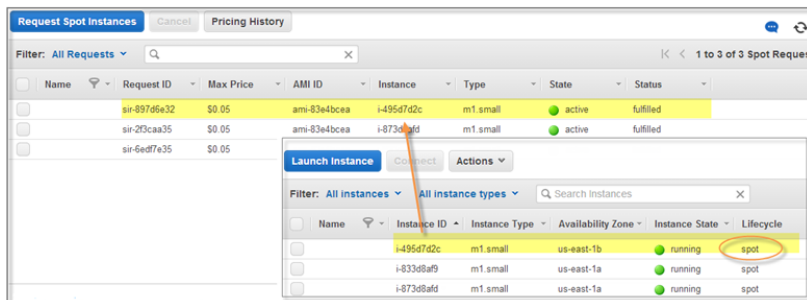
Amazon EC2 Command Line Interface (CLI) Tools

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.

```
$ 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. 606).

```
$   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                  ebs
     spot     sir-e1471206                  paravirtual
```

Finding Running Spot Instances Using the API

To get information about your running Spot Instances using the API, use the DescribeSpotInstanceRequests action.

What do you want to do next?

- Canceling Spot Instance Requests (p. 143)
- Launching Spot Instances in Amazon Virtual Private Cloud (p. 177)
- Tagging Spot Instance Requests (p. 156)
- 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. 134).
> Your Spot request can be in other states in addition to open and active. For more information, see Spot Instance Request States (p. 148). 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. 150).
> 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.
2. Select the Spot requests you want to cancel and click **Cancel**.
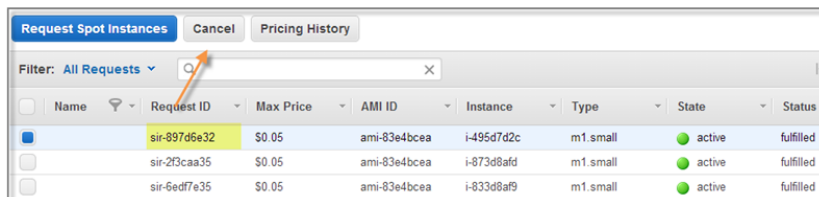
3. If you don't want the associated Spot Instances to continue running, you can cancel them. Select them in the **Instances** page, then right-click and select **Terminate**.
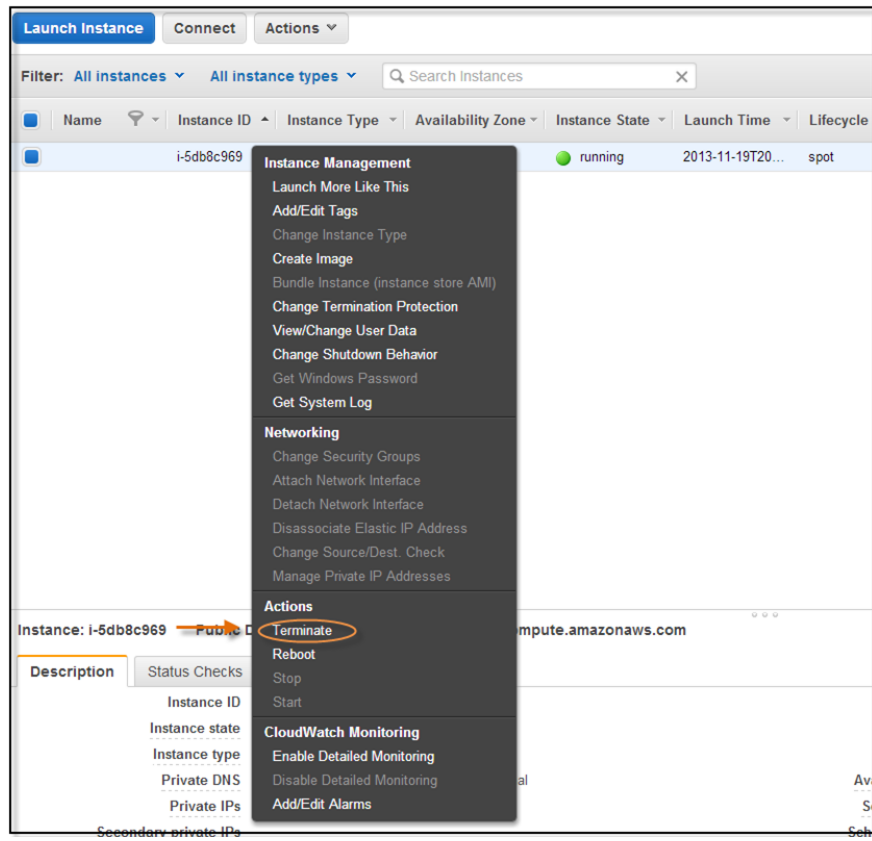
Amazon EC2 Command Line Interface (CLI) Tools

To cancel Spot Instance requests

1. Enter the following command to see your Spot Instance requests:

```
$ 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:

```
$ 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:

```
$ 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 EC2 Command Line Reference*.

Canceling Spot Instance Requests Using the API

To cancel Spot Instance requests using the API, use the CancelSpotInstanceRequests action.

What do you want to do next?

- Creating a Spot Instance Request (p. 138)
- Viewing Spot Instance Pricing History (p. 136)
- Finding Running Spot Instances (p. 141)
- Launching Spot Instances in Amazon Virtual Private Cloud (p. 177)

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. 146).
- You can use tags to manage your Spot Instances. For information, see Tagging Spot Instance Requests (p. 156).
- The interactions between Spot Price, available Spot capacity, and your bid price affect how long your Amazon EC2 Spot Instances run and when they get interrupted. For information, see Understanding Spot Instance Provisioning, Pricing, and Interruption (p. 157)
- 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).

For product and pricing information, see the following pages:

- Amazon EC2 Spot Instances
- AWS Service Pricing Overview

- Amazon EC2 On-Demand Instances Pricing

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.

> **Note**
> You cannot change the parameters of your Spot request, including the Spot price, after you've submitted the request.

This section discusses the bid fundamentals you need to know to help you manage your Spot requests.

- Amazon EC2 Spot Instance Limits (p. 146)

    Discusses the two categories of limits when using Spot Instances: Spot bid price limits and Spot Instance request limits per region.
- Customizing Your Spot Requests (p. 148)

    Discusses the options you can specify to customize your Spot request.
- Tracking Spot Requests with Bid Status Codes (p. 150)

    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.

For information about creating Spot Instance requests, see Creating a Spot Instance Request (p. 138).

Amazon EC2 Spot Instance Limits

There are two kinds of Spot Instance limits: bid price limits and request limits.

- A Spot *bid price* limit is the maximum price you can bid when you make a request for Spot Instances.
- A Spot Instance *request* limit is the number of instances per region that you can request.

For information about other AWS service limits, see AWS Service Limits.

Spot Bid Price Limits

On December 20, 2013, Amazon EC2 introduced a default limit on the maximum amount that you can bid for a Spot Instance. The default bid price limit is designed to protect Spot Instance customers from incurring unexpected charges due to high bid prices.

The default bid price limit is set to four times the On-Demand price. For example, if you submit a Spot Instance request for a Linux m3.2xlarge instance in the us-east-1 region, for which the On-Demand price is $0.900 per hour, you can bid up to $3.600 per hour.

If your application requires you to bid above the default bid price limit, you can submit a limit increase request to *spot-limit-increase@amazon.com*. Your limit increase request must include the following information:

* Your AWS account number.
* A description of how you use Spot Instances and manage Spot interruptions.
* The bid price limit you are requesting (specifying the instance type, region, and product platform).

For information about On-Demand instance pricing, see Amazon EC2 Pricing.

Spot Request Limits

The number of Spot Instance requests that an account can make per region varies depending on factors such as the region, the instance type, and the account's status. The default regional Spot request limit is generally 20 Spot Instances per account. However, your limit might be as low as 1 instance per region if you are a new AWS customer or if you are requesting certain EC2 instance types. In addition, the instance types T2, I2, and HS1 are not available on Spot, and some instance types, such as `g2.2xlarge`, are not available in all regions.

If you submit a Spot Instance request, and you receive an error stating `Max spot instance count exceeded`, your account is either at its overall Spot request limit for the region, or you are requesting a specific regional limit for an instance type. Try submitting another request with fewer instances.

The following table lists the different limits for certain instance types per region. Generally, accounts have a regional Spot request limit of 20.

Instance Type	Spot Limit
`cg1.4xlarge`	10
`g2.2xlarge`	10
`c3.4xlarge, c3.8xlarge`	20
`cc1.4xlarge`	20
`cc2.8xlarge`	20
`m2.2xlarge, m2.4xlarge`	20
`hi1.4xlarge`	10
`r3.large, r3.xlarge, r3.2xlarge`	10
`r3.4xlarge, r3.8xlarge`	5

For the Spot service to consider your Spot Instance request, you should not request a number of instances that exceeds your account's overall regional Spot request limit, or the instance type-specific Spot request limit.

For example, say you submit a Spot Instance request for 20 `m1.small` instances. Accounts generally are allowed 20 such instances per region. However, if your account is only allowed 10 `m1.small` instances per region, your request will not be considered. Alternatively, let's say that your account is allowed 100 instances per region and you submitted a Spot Instance request for 10 `r3.4xlarge` instances. The limit on this instance type is 5 per region. Your request will not be processed because you're requesting more `r3.4xlarge` instances than is allowed.

If you want to make additional Spot requests in a region, you can complete the Amazon EC2 instance request form. Specify in the **Use Case Description** that you are requesting an increase to your account's Spot Instance request limit. Limit increases apply to the region specified in the request form.

> **Note**
> Keep in mind that even if you are within your Spot request limits, your Spot Instances can be terminated if your bid price no longer exceeds the Spot price, or if there is a Spot capacity shortage.

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:

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. 150).

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 when the Spot price is $0.300, Amazon EC2 launches and keeps your instance running if your maximum bid price is above $0.300. If the Spot price rises above your maximum bid price and consequently your Spot Instance is terminated, your Spot Instance request remains active and Amazon EC2 will launch another Spot Instance for you when the Spot price falls below your maximum bid price.

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 Zone Groups

You can opt to have your Spot Instances launch at the same time or in the same Availability Zone. If you want Amazon EC2 to launch the instances in your Spot Instance request only if all of them can be fulfilled together, specify a *launch group* in your Spot Instance request. For more information, see Launch Groups (p. 149). If you want to launch your instances together in a single Availability Zone, specify an *Availability Zone group*. For more information, see Availability Zone Groups (p. 150).

You can specify these options using any of the following:

- The **Request Spot Instances** wizard in Amazon EC2 console
- The aws ec2 request-spot-instances AWS CLI command
- The ec2-request-spot-instances Amazon EC2 CLI command
- The RequestSpotInstances Amazon EC2 API action

Implement these options only when you need them. Although these options can be useful, they are requirements that can lower the chances that your Spot Instance requests are fulfilled.

Launch Groups

To tell Amazon EC2 to launch your instances only if all the instances in the Spot Instance request can be fulfilled, specify a launch group in your Spot Instance request using one of the following:

- The Amazon EC2 console
- The `--launch-group` option using the AWS CLI and Amazon EC2 commands
- The `LaunchGroup` option using the Amazon EC2 API action.

Keep in mind that specifying a launch group increases the chance that your Spot Instances will be terminated. When one instance in the launch group is terminated by the Spot service—for example, if the Spot price rises above your bid price—all of the instances in the group are terminated. If you submit another Spot Instance request specifying that the instances must be launched in the same (existing) launch group as an earlier successful request, then, if this request is successful, the instances in this new request will now be considered part of the launch group. Subsequently, if an instance in this launch group is terminated, all instances in the launch group will be terminated. And that includes instances launched by the first and second requests.

In addition, *Launch group* is a feature recognized and managed by the Spot service. If one or more instances in a launch group are terminated by you or a service other than Spot, the other instances in the launch group will not be terminated.

Availability Zone Groups

To tell Amazon EC2 to launch your instances in one Availability Zone, specify an **Availability Zone group** in your Spot Instance request using one of the following:

- The EC2 console
- The `--availability-zone-group` option using the AWS CLI and Amazon EC2 commands
- The `AvailabilityZoneGroup` option using the Amazon EC2 API action.

Before you use an Availability Zone group, you should know whether your account supports a default VPC (EC2-VPC). The way in which the Spot service places instances in Availability Zones differs depending on whether your account supports a default VPC (EC2-VPC). For more information about how Availability Zones are provisioned for Spot Instances, see Availability Zone Provisioning (p. 160). For more information, see Detecting Your Supported Platforms and Whether You Have a Default VPC.

The Spot service terminates instances in Availability Zones differently from the way it terminates instances in launch groups. When one instance in a launch group is terminated by the Spot service, all instances in the launch group will be terminated. In contrast, when one instance in an Availability Zone group is terminated, the other instances in the Availability Zone will remain running.

To identify a bid price for a Spot Instance in a specific Availability Zone, see Viewing Spot Instance Pricing History (p. 136) for the price over time in the Availability Zone where you want to make your bid. The most recent set of prices is 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.

The *describe Spot price history* commands return two pieces of data: the prices of the instance types within the time range that you specified and the time when the price changed, all within the specified Availability Zone group. The price is valid within the time period that you specified; the response merely indicates the last time that the price changed.

> **Note**
> If you use the Amazon EC2 API action (`DescribeSpotPriceHistory`) or the Amazon EC2 CLI command (`ec2-describe-spot-price-history`) 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.

When you review Spot prices, remember 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. 7).

Tracking Spot Requests with Bid Status Codes

Topics
- Life Cycle of a Spot Request (p. 151)
- Spot Bid Status Code Reference (p. 154)

Bid statuses can help you track your Amazon 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 Bottom pane, tooltip	Status code 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

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.
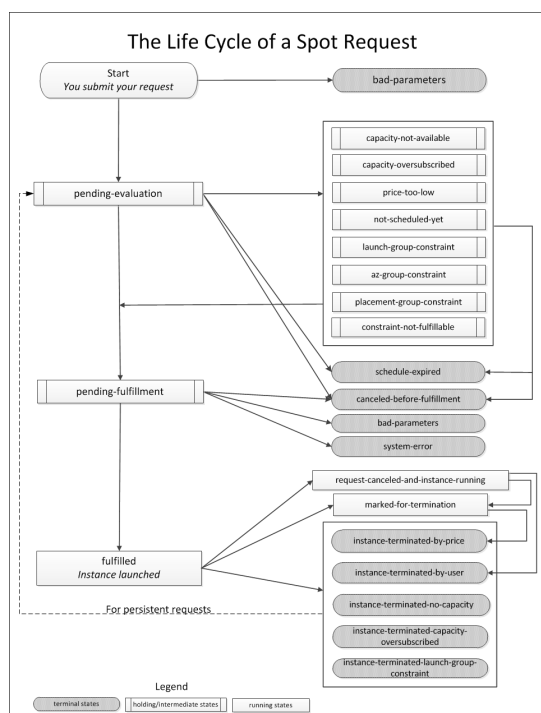
When you request a Spot Instance, your request goes through an evaluation process. At each step of the process—also called the Spot request *life cycle*—specific events dictate successive request states. In the life cycle diagram, each step is depicted as a node, and each node's status code describes the status of your Spot request and Spot Instance.

1. **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)

2. **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, 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)

3. **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 Price, your Spot request goes into the `pending-fulfillment` state.

Status Code	Request State	Instance State
`pending-fulfillment`	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
`fulfilled`	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 Price, Amazon 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. For more information, 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
`request-canceled-and-instance-running`	canceled[c]	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] canceled[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 might be a delay before the Spot service detects 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. With this status code, the request state can either be closed or canceled. Request state is *closed* if you terminate the instance, but keep your bid running. Request state is *canceled* if you terminate the instance and cancel the bid.

[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.

  > **Note**
  > 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 Zone Groups (p. 149).

- **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-and-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 assign the pair to the Spot Instance request.

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. When your request is fulfilled and a Spot Instance launches, the tag you used for your Spot Instance request is not applied to the Spot Instance. To categorize the Spot Instances that are launched from your tagged request, you must create a separate set of tags for the instances.

You can tag your Spot Instance request when you first create it, or you can assign a tag to the request after you create it. You can use the following tools to tag Spot Instance requests:

- The **Spot Requests** page of the EC2 console of the AWS Management Console
- create-tags AWS CLI command
- ec2-create-tags Amazon EC2 CLI command
- CreateTags API action

For information about tagging in Amazon EC2, see Tagging Your Amazon EC2 Resources (p. 610).

To create a tag for your Spot Instance request

1. When you create a Spot Instance request using the AWS Management Console, you first go through the steps of specifying your Amazon Machine Image (AMI) and instance type. For information, see Creating a Spot Instance Request (p. 138).

    After you have specified the instance configuration and storage details, click **Next: Tag Instances**.
2. In the **Tag Spot Request** step, type the key name and value you want for the tag, and click **Review and Launch**.

If you want to create another tag, click **Create Tag**.

3. If you want to create a tag for a Spot Instance request that already exists, in the **Spot Requests** page, select the Spot request that you want to tag, then, in the bottom pane, go to the **Tags** tab, and click **Add/Edit Tags**.

In the **Add/Edit Tags** box, click **Create Tag**, type the key name and value you want for the tag, and click **Save**.

If you want to apply tags to Spot Instances that are launched from the Spot request that you just tagged, you must create a separate set of tags for the instances using the same tools and commands. For information about tagging instances, see Tagging Your Amazon EC2 Resources (p. 610).

Understanding Spot Instance Provisioning, Pricing, and Interruption

If you have Amazon EC2 Spot Instances, you will want to know how your Spot Instances are placed in Availability Zones, how long your Spot Instances run, when they get interrupted, and how they are priced. These factors depend on the price you set for your bid, the Spot Price set by Amazon EC2, and the available Spot capacity. When you understand how these factors interact, you should be able to use Spot Instances more effectively.

This section discusses the following topics:

For pricing information, see Amazon EC2 Spot Instances.

Supply and Demand in the Spot Market

Spot Instances are spare EC2 instances made available at a price that varies with supply and demand. When you request a Spot Instance, you set the maximum price you are willing to pay (your bid). AWS continuously evaluates how much spare Amazon EC2 capacity in each instance pool can be offered on Spot, what bids have been made for each pool, and then provisions the available Spot capacity to the highest bidders. The Spot Price for the pool is set to the lowest fulfilled bid in that pool. (An instance pool is a group of instances that share the same region, Availability Zone, instance type, operating system, and network platform.) This means that, strictly speaking, the Spot Price is the price above which you have to bid to fulfill a Spot request for a single Spot Instance.

Here's an illustration of how this Spot provisioning and pricing mechanism works.

Assume that you are bidding on a Spot pool (e.g., the us-east-1a region and Availability Zone, m1.small instance type, Linux operating system, and VPC network platform) that has only five Spot Instances for sale. Your bid price is $0.100, which is also the current Spot Price. The current bids have the following bid prices.

> **Note**
> Bids are ranked in descending order. Bids that share the same bid price (tied bids) are ranked randomly.

Bid	Bid Price	
1.	$1.000	
2.	$1.000	
3.	$0.100	
4.	$0.100	Your bid
5.	$0.100	Last fulfilled bid sets the Spot Price
— — — —	— — — —	Spot capacity cutoff
6.	$0.050	

In the table, bids 1-5 are fulfilled. Bid 5, being the last fulfilled bid, sets the Spot Price at $0.100. Bid 6 is unfulfilled. Bids 3-5, which share the same bid price of $0.100, are stacked in random order.

Now, let's say that the spare Amazon EC2 capacity available to this Spot pool drops to 3. The Spot pool will now look like this:

Bid	Bid Price	
1.	$1.000	
2.	$1.000	
3.	$0.100	Last fulfilled bid sets the Spot Price
— — — —	— — — —	Spot capacity cutoff

Bid	Bid Price	
4.	$0.100	Your bid
5.	$0.100	
6.	$0.050	

Bids 1-3 are fulfilled. Bid 3, the last fulfilled bid, sets the Spot Price at $0.100. Bids 4-5, which also are $0.100, are unfulfilled. As you can see, even though the Spot Price didn't change, two of the $0.100 bids—Bid 4 (your bid) and Bid 5—have been interrupted because the Spot supply shrank.

If you still want to fulfill a Spot request for a single instance, you would have to bid above the current Spot Price of $0.100. If you bid $0.101, your Spot Instance would be fulfilled, Bid 3 would be interrupted and the Spot Price would become $0.101. If you bid $2.000, Bid 3 would be interrupted and the Spot Price would become $1.000 (Bid 2's price). Keep in mind that, in the scenario above, no matter how high you bid, you could never get more than 3 instances fulfilled.

If spare Amazon EC2 capacity available to Spot dropped to zero, all bids would be interrupted.

> **Note**
> This means that your Spot Instance could be interrupted, no matter how high you bid. Never bid higher than you're willing to pay because, when there is price competition in the Spot market—typically for a small amount of available Spot capacity—Spot Prices can rise unexpectedly and remain elevated for extended periods of time.

Understanding Interruptions

When you use Spot Instances, you must always be prepared for interruptions. Your instances can be interrupted at any time; they are not protected by a high Spot bid. Consequently, you must ensure that you have a backup system for your data. Instance data is not saved by default; it will be lost when the instance is interrupted.

There are three possible reasons why your Spot Instances get interrupted.

* Spot Price—If your bid price, which is the maximum price you're willing to pay for a Spot Instance, is lower than the current Spot Price, your running Spot Instance will be terminated.

  Amazon EC2 sets a Spot Price for each instance type in each Availability Zone. The Spot Price is the price all customers will pay to run a Spot Instance for that given period. The Spot Price fluctuates based on supply and demand for instances, but customers will never pay more than the maximum price that they have specified in their Spot request. If the Spot Price moves higher than a customer's maximum price, the customer's instance will be shut down by Amazon EC2.
* Capacity—If there is not enough spare Amazon EC2 capacity, Spot Instances will be terminated, starting with those instances with the lowest bids. If there are several instances with the same bid price, the order of instance termination is determined at random.
* Request constraints—If you requested Spot Instances with constraints, such as the same launch group or the same Availability Zone group, these instances can be interrupted as a group if the constraint can no longer be met.

The cause of your Spot Instance's interruption affects how you are charged. If you terminate your Spot Instance, you are charged for a full hour of usage regardless of whether you used it a full hour. If Amazon EC2 terminates your Spot Instance, you are not charged for the terminated hour. Your account is not charged until your request for a new Spot Instance is placed and your Spot Instance is launched.

Availability Zone Provisioning

When you submit a request for Spot Instances and your bid price is higher than the Spot price in the Availability Zone you specified, your instances will be launched in your chosen zone as long as there's capacity for them. However, when you don't specify the Availability Zone in your Spot Instance request, how the Spot service manages your instances when the request succeeds depends on whether your account supports a default VPC (EC2-VPC) or supports EC2-Classic.

EC2-Classic

If you have an EC2-Classic account, which does not support a default VPC, and you request to launch all of your Spot Instances in an Availability Zone group without identifying a specific Availability Zone, Amazon EC2 will find the lowest-priced Availability Zone in the region and attempt to launch all your Spot Instances in that zone, assuming that the price of the lowest-priced bid in the group is higher than the current Spot price in that zone. Amazon EC2 will keep on trying until there is enough capacity to launch your Spot Instances together, as long as the Spot price remains lower than the lowest price in the group.

EC2-VPC

If your account supports a default VPC (EC2-VPC), you can't request to launch all your Spot Instances in the cheapest Availability Zone. The choice of the Availability Zone depends on certain conditions. Assuming that the Spot price remains lower than the lowest bid price in the group and there is enough capacity in the selected Availability Zone, one of the following will occur:

- If your account has a default VPC and subnet, and they're available, then Amazon EC2 will attempt to launch your Spot Instances together in the default subnet (and hence in the Availability Zone that the subnet belongs to).
- If your account has a default VPC and subnet, and they were previously removed, then you will be required to specify the subnet into which you want to launch all the instances.
- If your account has a default VPC and subnet, and you tell Amazon EC2 to place your instances in another subnet-Availability Zone, and it exists, then Amazon EC2 will attempt to launch your Spot Instances together in the subnet that you specified.
- If your account has a default VPC and subnet, and you tell Amazon EC2 to place your instances in another subnet and Availability Zone that don't exist, then you will get an InvalidParameterValue exception. For more information, see *InvalidParameterValue* in Client Error Codes
- If your account has no default VPC and subnet, then Amazon EC2 will create a VPC and subnets in all Availability Zones for your account and attempt to launch your Spot Instances together in the default subnet.

For more information, see Detecting Your Supported Platforms and Whether You Have a Default VPC.

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. In addition, no matter how high you bid, there is always a risk that your Spot Instance will be interrupted. Thus, 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. We strongly recommend against bidding above the On-Demand price or using Spot for applications that cannot tolerate interruptions.

> **Note**
> You can use AMIs backed by Amazon Elastic Block Store, but you cannot stop and start EBS-backed instances if they are Spot Instances. Spot Instances are terminated if the Spot price rises above the bid price that launched the instance, and the Spot service does not support restarting instances that have been stopped. For information about Amazon EBS, see Amazon Elastic Block Store (Amazon EBS) (p. 519).

Planning for Interruptions

Because Spot Instances can terminate at any time, applications that run on Spot Instances must terminate cleanly. Although we attempt to terminate your instances cleanly, 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. 173)
- Starting Clusters on Spot Instances (p. 183)

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. 177)

You can use AWS services with Spot Instances. In this section, we will show you how Amazon EC2 Spot Instances work with other services, such as Auto Scaling, Amazon 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. 163)
- Obtaining Information About the Instances Launched by Auto Scaling (p. 166)
- Updating the Bid Price for the Spot Instances (p. 170)
- Scheduling Spot Bid Requests (p. 173)
- Using Auto Scaling to Get Notifications for Spot Instances (p. 173)

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 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 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 Price: If the bid price is higher than the Spot Price, then an instance will be launched; if the Spot 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, see Auto Scaling 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 *Auto Scaling Developer Guide*.
Read the readme installation instructions	*Install_Drive\AutoScaling-n.n.nn.n*\README

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
`as-create-launch-config`	Creates a new launch configuration with specified attributes.
`as-create-auto-scaling-group`	Creates a new Auto Scaling group with specified name and other attributes.
`as-describe-auto-scaling-groups`	Describes the Auto Scaling groups, if the groups exist.
`as-describe-auto-scaling-instances`	Describes the Auto Scaling instances, if the instances exist.
`as-describe-scaling-activities`	Describes a set of activities or all activities belonging to a group.
`as-delete-auto-scaling-group`	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 EC2 Command Line Reference*.
- **Amazon EC2 CLI tools specific to Spot Instances.** Using the AWS CLI and the EC2 CLI Tools for Spot Instances (p. 135).
- **Other Amazon EC2 tools you can use with Spot Instances.** Getting Started with Spot Instances (p. 133).

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 EC2 User Guide for Linux Instances*. 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. 170)
- Scheduling Spot Bid Requests (p. 173)
- Using Auto Scaling to Get Notifications for Spot Instances (p. 173)
- Advanced Tasks (p. 180)

For more information about Auto Scaling, see What is Auto Scaling? 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 Instance Pricing 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-dis-
abled] [--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 Finding a Linux AMI (p. 58).

- 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 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: `as-create-auto-scaling-group --help`. For more information, see Resources for Using Auto Scaling CLI Tools (p. 162).

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. 166)
- Updating the Bid Price for the Spot Instances (p. 170)
- Scheduling Spot Bid Requests (p. 173)
- Using Auto Scaling to Get Notifications for Spot Instances (p. 173)
- Advanced Tasks (p. 180)

For more information about Auto Scaling, see What is Auto Scaling? 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   ACTIVITY-ID                               END-TIME              GROUP-
    NAME   CODE                          MESSAGE
    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
    ```

    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 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   ACTIVITY-ID                               END-TIME              GROUP-
    NAME       CODE
    ACTIVITY   90630906-b40f-41a6-967a-cd6534b2dfca  2012-06-01T02:32:15Z  spotasg
             Successful
    ACTIVITY   a1139948-ad0c-4600-9efe-9dab8ce23615  2012-06-01T00:48:02Z  spotasg
             Successful
    ACTIVITY   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>
        <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-fe30d187`

  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 Price exceeded their bid price.

  When Auto Scaling attempts to replace terminated instances resulting from the Spot 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 shows 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      STATUS    LAUNCH-CONFIG
INSTANCE   i-2c30d155    us-east-1a          InService  Healthy   spotlc-5cents
INSTANCE   i-fe30d187    us-east-1a          InService  Healthy   spotlc-5cents
INSTANCE   i-c630d1bf    us-east-1a          InService  Healthy   spotlc-5cents
```

You can see that Auto Scaling launched three 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 returns 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. 170)
   - Scheduling Spot Bid Requests (p. 173)
   - Using Auto Scaling to Get Notifications for Spot Instances (p. 173)
   - 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 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 Finding a Linux AMI (p. 58).

- 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 --de
sired-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-7a24-4aaf-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 Price goes above the bid price on which it was based.

What do you want to do next?

- 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 `as-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. 166)
- Updating the Bid Price for the Spot Instances (p. 170)
- Using Auto Scaling to Get Notifications for Spot Instances (p. 173)
- 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 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, see 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, see 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, see PutNotificationConfiguration in the *Auto Scaling API Reference*.

A list of Auto Scaling notification types. These notification types are the events that 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, see 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 AWS resources. Replace the ARN placeholder with your ARN.

- Notification types: `autoscaling:EC2_Instance_Launch`, `autoscaling:EC2_Instance_Ter-minate`

  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_IN
STANCE_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_IN
STANCE_LAUNCH
NOTIFICATION-CONFIG   spotasg   arn:placeholder:spotasg   autoscaling:EC2_IN
STANCE_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, Amazon 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. 166)
- Updating the Bid Price for the Spot Instances (p. 170)
- 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 a *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 AWS CloudFormation templates utilize Spot Instances, and you can download them from the template library:

- 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.

- 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; 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.

- 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.

You can create stacks from these templates by using the AWS Management Console, the AWS Cloud-Formation 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, see Getting Started with AWS CloudFormation in 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 EMR 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 VPC 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 a VPC. To ensure that your Spot Instances are launched in a 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, determine whether you want to launch your instances in that VPC. You don't have to perform any additional steps when you request Spot Instances. The **Request Spot Instances** wizard lists the VPC ID in the **Network** field.
- If your AWS account supports a default VPC, determine whether you want to use a different VPC for your Spot Instances. You can create a new VPC and specify the ID of the VPC that you just created.
- If your AWS account does not support a default VPC, create a VPC and specify the ID of the VPC that you just created.

For more information about default VPCs, see Default VPC Basics. For more information about setting up a VPC, see Getting Started with Amazon VPC in the *Amazon VPC Getting Started Guide*.

Quick Look: Launching Spot Instances in Amazon VPC Video

The following video shows how to launch your first Spot Instance in Amazon VPC using the AWS Management Console. This video includes instructions for creating your VPC subnet, placing a bid, determining when the instance is fulfilled, and canceling the instance. Launching Spot Instances in Amazon VPC Video

Understanding the Process for Launching Spot Instances in Amazon VPC

The process for making a Spot Instance request that launches in a VPC is the same as the process you follow when you make a Spot Instance request in a non-VPC portion of Amazon EC2—except for the following differences:

You should base your bid on the Spot Price history of Spot Instances in VPCs.

When you use the describe-spot-price-history AWS CLI command or the ec2-describe-spot-price-history Amazon EC2 CLI command, add *(Amazon VPC)* to the `product-description` filter.

Using the AWS CLI, here's how the command looks:

```
$ aws ec2 describe-spot-price-history --instance-types m1.xlarge --product-de
scription "Linux/UNIX (Amazon VPC)" --start-time 2014-01-06T07:08:09 --end-time
 2014-01-06T08:09:10
```

Using the AWS Management Console, check the **Spot Instance Pricing History** page to see the Spot pricing history for Amazon EC2 instances running in both Amazon EC2 and Amazon VPC.
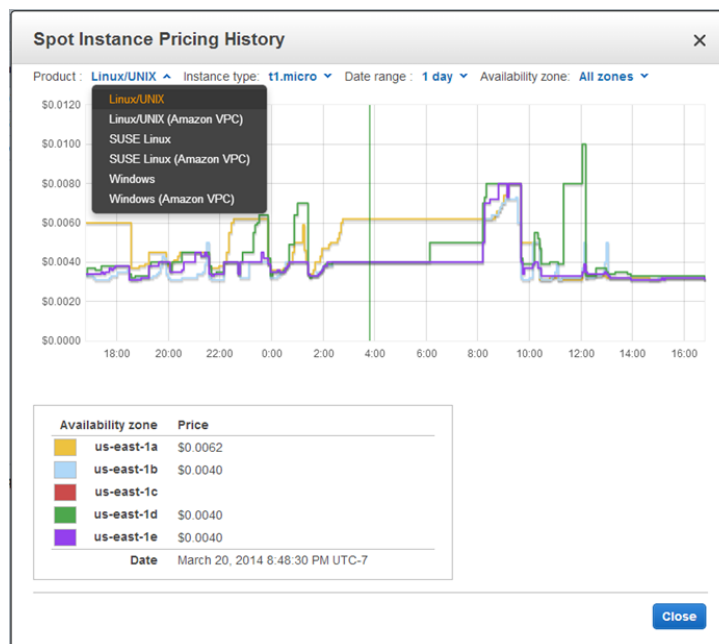
You have to specify the VPC in which you want to launch your Spot Instance.

When you use the aws ec2 request-spot-instances AWS CLI command or the ec2-request-spot-instances Amazon EC2 CLI command, specify the ID of the VPC subnet in which you want to launch the Spot Instance.

Using AWS CLI, here's how the command looks:

```
$ aws ec2 request-spot-instances --spot-price "0.050" --instance-count 5 --type
 "one-time" --launch-specification "{\"ImageId\":\"ami-a43909e1\",\"Instance
Type\":
  "m1.small\",\"SubnetId\":\"subnet-d50bfebc\"}"
```

Using the AWS Management Console, when you launch the **Request Spot Instances** wizard from the **Spot Requests** page of the Amazon EC2 console, in the **Configure Instance Details** page, select a VPC from the **Network** drop-down. What the **Network** displays depends on whether or not your account supports a default VPC. (For more information, see Detecting Your Supported Platforms and Whether You Have a Default VPC.)

- If your account is EC2-Classic and therefore does not support a default VPC, **Network** will show VPC options if you have a VPC before making the Spot Request. For example, if you are requesting Spot Instances in us-east-1 and you have VPCs, **Network** will list the VPC subnets available for the region's Availability Zones.
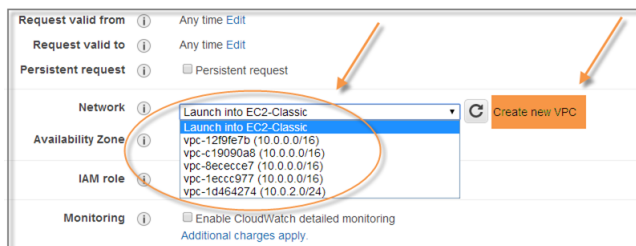
If you don't have a VPC, click **Create new VPC**. The **VPC Management Console** opens in a new tab. Click **Create VPC** and follow the wizard to create a new VPC.

- If your account supports a default VPC, **Network** will list the default VPC. You will also see a **Subnet** drop-down list that shows the default subnets for the Availability Zones in the region in which you're requesting your Spot Instances. You can opt to create a new VPC or a new subnet.
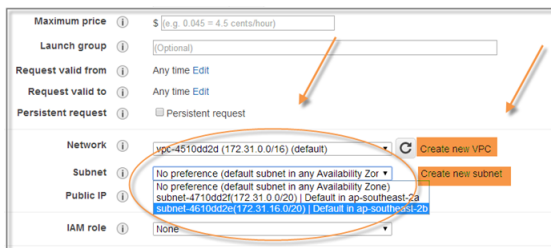
If you want your Spot Instance launched in a specific low-priced Availability Zone, you must specify the subnet in your Spot Instance request.

The subnet you choose also determines the Availability Zone in which your Spot Instance will be launched. If you do not specify a subnet and Availability Zone when you submit your Spot Instance request, your Spot Instance will be placed in the subnet of an available Availability Zone. The Availability Zone might not have the lowest Spot price.

If your account supports a default VPC and you do not specify a subnet and Availability Zone in your Spot request, the default VPC that the placement system chooses might be different from the Availability Zone with the lowest Spot price.

For more information about using Amazon VPC, see the Amazon VPC User Guide.

Advanced Tasks

Now that you have created Spot Instance requests and worked with Spot Instances, this section discusses some advanced tasks.

If you think we should add other advanced tasks to this section, let us know at *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 Amazon 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:

```
$  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.amazon
aws.com/doc/2014-10-01/">
  <requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId>
  <spotDatafeedSubscription>
       <ownerId>999988887777</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:

```
$  ec2-delete-spot-datafeed-subscription
```

If the request is successful, the output is empty.

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.amazon
aws.com/doc/2014-10-01/">
    <requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId>
    <return>true</return>
</DeleteSpotDatafeedSubscriptionResponse>
```

Programming Spot Instances the with AWS Java SDK

You can use the AWS Java SDK to program Spot Instances. Try the following tutorials to get started:

- Tutorial: Amazon EC2 Spot Instances
- Tutorial: Advanced Amazon EC2 Spot Request Management

This table lists information that might be useful as you use the AWS Java SDK to program Spot Instances and Amazon EC2 in general.

To find information about:	See this documentation:
Java basics	AWS SDK for Java Basics
Using Amazon EC2 with Java	Programming Amazon EC2 with the AWS SDK for Java
Java API Reference	AWS SDK for Java API Reference

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

You can use Amazon Elastic Compute Cloud (Amazon EC2) Reserved Instances to pay a one-time fee to reserve capacity and receive a discount on your instance use compared to running On-Demand instances. The discounted usage price is locked for the duration of your contract, allowing you to predict compute costs over the term of the Reserved Instance. When you want to use the capacity you reserved, you launch an EC2 instance—the same way you launch an On-Demand instance—with the same configuration as the reserved capacity that you purchased. Amazon Web Services (AWS) will automatically apply the usage price that is associated with your capacity reservation. You are charged that usage price for your EC2 instance for as long as you own the Reserved Instance.

When the term of your Reserved Instance ends, and you do not renew by purchasing another capacity reservation, you can continue using the EC2 instance without interruption. However, you will now be charged at the On-Demand rate.

To purchase an Amazon EC2 Reserved Instance, you must select an instance type (such as m1.small), platform (Linux/UNIX, Windows), location (region and Availability Zone), and term (either one-year or three-year). You can use Auto Scaling or other AWS services to launch the On-Demand instances that utilize your Reserved Instance benefits. For information about launching On-Demand instances, see Launch Your Instance (p. 289). For information about launching instances using Auto Scaling, see the Auto Scaling Developer Guide.

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 reservation and pricing benefit. Both your Reserved Instance purchase and the instance you launch must specify the same *SUSE Linux* product platform.

For product pricing information, see the following pages:

* AWS Service Pricing Overview
* Amazon EC2 On-Demand Instances Pricing
* Amazon EC2 Reserved Instance Pricing

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. 186).
* Before you buy and sell Reserved Instances, you can learn more about the process by reading Steps for Using Reserved Instances (p. 187).
* 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. 191).
* Learn more about the pricing benefit of Reserved Instances. For more information, see Understanding the Pricing Benefit of Reserved Instances (p. 198).

- Understand Reserved Instance pricing tiers and how to take advantage of discount pricing. For more information, see Understanding Reserved Instance Pricing Tiers (p. 193).
- 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. 200).

For a checklist that summarizes requirements for working with Reserved Instances and the Reserved Instance Marketplace, see Requirements Checklist for Reserved Instances (p. 254).

What Do You Want to Do Next?

Getting Started with Reserved Instances

Topics

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. Compared to running On-Demand instances, with Reserved Instances you pay a low, one-time fee and in turn receive a significant discount on your instance usage during the term of your Reserved Instance. The discounted usage price is locked for the duration of your contract, allowing you to predict compute costs over the term of your Reserved 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 basic information you need to get started with Reserved Instances

Setting 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. 26), which provides information on signing up for your Amazon EC2 account and credentials.

* Install the tools.

  You can use the AWS Management Console, command line interface tools, the Amazon EC2 API, as well as the AWS SDK to work with EC2 Reserved Instances and search for offerings. For more information, see Tools for Working with Reserved Instances (p. 189).

  To start working right away with Reserved Instances using specific tools, see:
  * AWS Management Console (p. 189)
  * Command Line Interface Tools (p. 190)
  * API (p. 190)

Steps for Using Reserved Instances

There are five sets of steps to follow when you use Reserved Instances. You can *purchase* Amazon EC2 Reserved Instances, and then you can *use* them by launching On-Demand instances with matching specifications. You can *view* the Reserved Instances you have, *modify* them, and you can *sell* unused Reserved Instances in the Reserved Instance Marketplace. (Restrictions apply. For information, see Requirements Checklist for Reserved Instances (p. 254).) This section describes purchasing, using, viewing, modifying, 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 Setting Up (p. 186).

> **Note**
> To purchase and modify Reserved Instances, ensure that your account has the appropriate permissions, such as the ability to describe Availability Zones. For information, see the IAM Best Practices and the Permissions and Policies sections in the *Using IAM* guide.

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. 191).

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. 205).

2. Use.

To use your Reserved Instance, launch an On-Demand EC2 instance with the same criteria as your Reserved Instance—the same region, Availability Zone, instance type, and platform specified when you purchased your Reserved Instance. Reserved Instance pricing benefits and capacity reservations automatically apply to any matching EC2 instances you have that aren't already covered by a reservation. Amazon Web Services (AWS) will apply the usage price that is associated with your capacity reservation. The usage price for your EC2 instance remains the same for as long as you own the Reserved Instance.

For more information, see Launch Your Instance (p. 289).

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. 213).

4. Modify.

You can modify your Reserved Instances by moving them between Availability Zones within the same region, changing their instance type to another instance type in the same instance family (e.g., the M1 instance family), or modifying the network platform of the Reserved Instances between EC2-VPC and EC2-Classic.

For more information, see Modifying Your Reserved Instances (p. 218).

5. 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. 229).

> **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. 254).

b. Decide on a price for the Reserved Instances that you want to sell. For more information, see Pricing Your Reserved Instances (p. 234).

c. List your Reserved Instances. For more information, see Listing Your Reserved Instance (p. 234).

d. Find out how you get paid when your Reserved Instances are sold. For more information, see Getting Paid (p. 254).

Using Reserved Instances in Amazon VPC

To launch instances that utilize Reserved Instances benefits 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 VPC 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 reservation and billing benefits are applied to your instance. For information about default VPCs, see Your Default VPC and Subnets in the *Amazon VPC 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 VPC User Guide*.

Tools for Working with Reserved Instances

You can use the AWS Management Console (p. 189), the Command Line Interface Tools (p. 190), or the API (p. 190) to list or search for available Amazon EC2 Reserved Instances, purchase reserved capacity, manage your Reserved Instance, and sell your unused Reserved Instances.

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.

- Use the **Purchase Reserved Instances** page to specify the details of the Reserved Instances you want to purchase.

- Use the **Instances** page to launch and manage instances that utilize your Reserved Instance benefits.

Command Line Interface Tools

To purchase Reserved Instances, or sell them in the Reserved Instance Marketplace, you can use the AWS Command Line Interface (AWS CLI) or the EC2 command line interface (Amazon EC2 CLI) tools specifically designed for these tasks. To manage the instances for your Reserved Instances, use the same commands in the CLI tools that you use for any other Amazon EC2 instances.

Before you use these tools, read and complete the following instructions:

- **AWS CLI:** Read and complete the instructions described in Getting Set Up with the AWS CLI Tools. The Getting Started topic walks you through setting up your environment for use with the AWS CLI tools.
- **Amazon EC2 CLI:** Read and complete the instructions described in Setting Up the Amazon EC2 Tools. The Getting Started topic walks you through setting up your environment for use with the EC2 CLI tools.

The following table lists the commands in the CLI tools that you use specifically for Reserved Instances tasks.

Task	AWS CLI	Amazon EC2 CLI
List Reserved Instances that you have purchased	`aws ec2 describe-reserved-instances`	`ec2-describe-reserved-instances`
Modify the Reserved Instances you own	`aws ec2 modify-reserved-instances`	`ec2-modify-reserved-instances`
View the modifications made to your Reserved Instances	`aws ec2 describe-reserved-instances-modifications`	`ec2-describe-reserved-instances-modifications`
View the Reserved Instances offerings that are available for purchase	`aws ec2 describe-reserved-instances-offerings`	`ec2-describe-reserved-instances-offerings`
Create a listing of the Reserved Instances you want to sell in the Reserved Instance Marketplace	`aws ec2 create-reserved-instances-listing`	`ec2-create-reserved-instances-listing`
View the details of your Reserved Instance listings in the Reserved Instance Marketplace	`aws ec2 describe-reserved-instances-listings`	`ec2-describe-reserved-instances-listings`
Purchase a Reserved Instance	`aws ec2 purchase-reserved-instances-offering`	`ec2-purchase-reserved-instances-offering`
Cancel your active Reserved Instances listing in the Reserved Instance Marketplace	`aws ec2 cancel-reserved-instances-listing`	`ec2-cancel-reserved-instances-listing`

For information about AWS CLI commands, see the AWS Command Line Interface Reference. For information about EC2 CLI commands, see the Amazon EC2 Command Line Reference.

API

To purchase Reserved Instances, or sell them in the Reserved Instance Marketplace, you can use the API calls specifically designed for these tasks. To manage the instances for your Reserved Instances, 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.	`DescribeReservedInstances`
Modify the Reserved Instances you own.	`ModifyReservedInstances`
View the modifications made to your Reserved Instances.	`DescribeReservedInstancesModifications`
View the Reserved Instances offerings that are available for purchase.	`DescribeReservedInstancesOfferings`
Create a listing of the Reserved Instances you want to sell in the Reserved Instance Marketplace.	`CreateReservedInstancesListing`
View the details of your Reserved Instance listings in the Reserved Instance Marketplace.	`DescribeReservedInstancesListings`
Purchase a Reserved Instance.	`PurchaseReservedInstancesOffering`
Cancel your active Reserved Instances listing in the Reserved Instance Marketplace.	`CancelReservedInstancesListing`

For information about API actions, see the Amazon EC2 API Reference.

Reserved Instance Fundamentals

This section discusses fundamental concepts that can help you optimize the benefits of Reserved Instances, and use and manage them effectively.

- Choosing Reserved Instances Based on Your Usage Plans (p. 191)—Select the pricing model that best matches how often you plan to use your instances.
- Understanding Reserved Instance Pricing Tiers (p. 193)—Take advantage of the Reserved Instances pricing tier discounts in a region, when the total upfront list price of your Reserved Instances in the region is $250,000 USD or more.
- Understanding the Pricing Benefit of Reserved Instances (p. 198)—Learn how the Reserved Instances pricing benefits are applied.
- Reserved Instance Marketplace (p. 200)—Understand the flexibility provided by the Reserved Instance Marketplace to AWS customers who can sell the remainder of their Reserved Instances when their needs change, or buy Reserved Instances with less than the full standard terms from other AWS customers.

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. 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.

Understanding Reserved Instance Pricing Tiers

To qualify for Amazon Elastic Compute Cloud (EC2) Reserved Instances pricing tier discounts in a region, the total upfront list price of your Reserved Instances in the region must be $250,000 USD or more. When your account qualifies for a discount pricing tier, it will automatically receive discounts on upfront and usage fees for all Reserved Instance purchases that you make within that tier level from that point on.

This section introduces you to Reserved Instances pricing tiers and how to take advantage of the pricing tier discounts.

- What Are the Reserved Instance Pricing Tiers? (p. 193)
- Current Limitations (p. 194)
- Determining Your Pricing Tier Level (p. 194)
- How Do Pricing Tier Discounts Get Applied? (p. 196)

For information about how the pricing benefit of Reserved Instances is applied, see Understanding the Pricing Benefit of Reserved Instances (p. 198). For information about Reserved Instances pricing, see Amazon EC2 Reserved Instance Pricing.

What Are the Reserved Instance Pricing Tiers?

The following table lists the qualifications for each pricing tier and the discount that is applied when the total upfront list price of your active Reserved Instances in a region crosses into the range of that pricing tier.

Tier level	Total upfront list price of active Reserved Instances in the region	Discount applied to upfront and usage fees for Reserved Instances purchased in the tier
Tier 0	$0 - $249,999	Standard Reserved Instance upfront and usage fees. No discount.
Tier 1	$250,000 - $1,999,999	5 percent discount
Tier 2	$2,000,000 - $5,000,000	10 percent discount
Tier 3	Over $5,000,000	Contact Us

Current Limitations

The following limitations currently apply to Reserved Instances pricing tiers:

- Amazon EC2 Reserved Instance purchases are the only purchases that will apply toward your Amazon EC2 Reserved Instance pricing tier discounts. 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 Reserved Instances for Windows with SQL Server Standard or Windows with SQL Server Web.
- 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. 200).

For a checklist that summarizes requirements for working with Reserved Instances and the Reserved Instance Marketplace, see Requirements Checklist for Reserved Instances (p. 254).

Determining Your Pricing Tier Level

To determine which Reserved Instance pricing tier applies to you in a particular region, compare the sum of your Reserved Instances' *upfront* list prices to the total upfront list price required for the pricing tier. *List price* is the undiscounted Reserved Instance price that you see in the AWS Management Console or the AWS marketing website. Keep in mind that if the price of Reserved Instances drops after you buy Reserved Instances, that price drop might not be reflected in the undiscounted Reserved Instance price of your Reserved Instances.

List price is not the same as paid price. *Paid price* is the actual amount that you paid for the Reserved Instances. (The term *paid price* is the same as the term *fixed price* that you see when you use Reserved Instance tools in the AWS Management Console, command line interface (CLI), or the API.) If the Reserved Instance was purchased at a discount, the paid price will be lower than the list price. If the Reserved Instance was purchased without a discount, the paid price will equal the list price. So, if you purchased your Reserved Instances without discounts, you can use the fixed price value in the console, the CLI, or the API to determine which pricing tier your purchase falls under.

> **Note**
> In the Amazon EC2 console, you might need to turn on the display of the **Fixed Price** column by clicking **Show/Hide** in the top right corner.

Assuming you purchased your Reserved Instances without discounts, you can determine the pricing tier for your account by calculating the total fixed price for all your Reserved Instances in a region. To do this using the console, calculate the sum of the amounts in the **Fixed Price** column.

Using the AWS CLI, Amazon EC2 CLI, or the Amazon EC2 API, you can determine the pricing tier of your account by calculating the sum of the *fixed price* values returned by the `aws ec2 describe-reserved-instances` AWS CLI command, the `ec2-describe-reserved-instances` EC2 command, or the `DescribeReservedInstances` action, respectively.

Your AWS CLI command should look like the following example:

```
$ aws ec2 describe-reserved-instances
```

Amazon EC2 returns output similar to the following example:

```
{
    "ReservedInstances": [
        {
            "ReservedInstancesId": "b847fa93-e282-4f55-b59a-1342fexample",
            "OfferingType": "Medium Utilization",
            "AvailabilityZone": "us-west-1a",
            "End": "2016-08-14T21:34:34.000Z",
            "ProductDescription": "Linux/UNIX",
            "UsagePrice": 0.01,
            "RecurringCharges": [],
            "Start": "2013-08-15T21:34:35.086Z",
            "State": "active",
            "FixedPrice": 82.0,
            "CurrencyCode": "USD",
            "Duration": 94608000,
            "InstanceTenancy": "default",
            "InstanceType": "t1.micro",
            "InstanceCount": 10
        },
        {
            "ReservedInstancesId": "d16f7a91-d8b4-4b0f-bef9-aef08example",
            "OfferingType": "Medium Utilization",
            "AvailabilityZone": "us-west-1c",
            "End": "2016-08-14T21:34:34.000Z",
            "ProductDescription": "Linux/UNIX",
            "UsagePrice": 0.01,
            "RecurringCharges": [],
            "Start": "2013-08-15T21:34:35.455Z",
            "State": "active",
            "FixedPrice": 82.0,
            "CurrencyCode": "USD",
            "Duration": 94608000,
            "InstanceTenancy": "default",
            "InstanceType": "t1.micro",
            "InstanceCount": 10
```

```
        },
        ...
      ]
}
```

For an example using the Amazon EC2 CLI command, see ec2-describe-reserved-instances. For an example using the Amazon EC2 API action, see DescribeReservedInstances.

How Do Pricing Tier Discounts Get Applied?

If a single purchase of Reserved Instances in a region 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 determine 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 that shows you the discount effect of a purchase of Reserved Instances that crosses the discount tier threshold. 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 $1,000 each. That's a total of $75,000, which brings the total amount you have paid for active Reserved Instances to $275,000. Because 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 5 percent ($1,250). This means you will only be charged $23,750 for the remainder of your purchase (25 instances), and you will be charged discounted usage rates for those 25 Reserved Instances. (However, keep in mind that your total upfront— *undiscounted* —list price is still $275,000.)

After the total upfront list price of your active Reserved Instances in a region crosses into the discount pricing tier, any future purchase of Reserved Instances in that region will be charged at a discounted rate. As long as your total list price stays above the price point for the discount tier, all future purchases of Reserved Instances in that region will be discounted. If your total list price falls below that price point for the discount tier—for example, if some of your Reserved Instances expire—succeeding purchases of Reserved Instances in the region will not be discounted. However, you will continue to get the discount against the already purchased Reserved Instances that originally were within the discount pricing tier.

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 account reaches the discount tier level, any Reserved Instances purchased after this point by any 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 the discount tier 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 5 percent. So, when account B purchases Reserved Instances at a list price of $15,000, the consolidated account will only be charged $14,250 for the new Reserved Instances ($15,000 minus the 5 percent discount of $750 equals $14,250), 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. 199).

Purchasing at a Discount Tier Price

When you purchase Reserved Instances, Amazon EC2 will automatically apply any discounts to the part of your Reserved Instance purchase that falls within a discount 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.
- AWS CLI: Use the `aws ec2 purchase-reserved-instances-offering` command.
- 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 console, the `aws ec2 describe-reserved-instances` and `ec2-describe-reserved-instances` commands, 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.

Consequently, the Reserved Instance ID returned by your purchase CLI command or API action will be different from the actual ID of the new Reserved Instances. For more information, see Reserved Instance IDs (p. 198).

For example, let's say that your account has 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 5 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 console, the CLI, or the API will show two reservations—a reservation for $5,000 at the undiscounted rate, and another for $4,750, which is the discounted price. The discount is 5 percent of $5,000 or $250.

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 $254,750 in Reserved Instances. $250,000 is at the undiscounted rate, and $4,750 is at the discounted rate. In two months, Reserved Instances worth $20,000 expire, bringing your total down to $234,750, which is below the threshold for the 5 percent discount tier. When you subsequently 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 in the console, the AWS CLI, Amazon EC2 CLI, and the Amazon EC2 API is the *paid price* of all Reserved Instances purchased. The discount tier is based on *list price*.

Remember that when you purchase Reserved Instances, one of four possible scenarios will occur:

- Your purchase of Reserved Instances within a region is still below the discount threshold. This means that 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 when you use the purchase command will be different from the new Reserved Instance IDs that will actually be created at the completion of the purchase. These IDs are returned when you use the describe command. For an explanation of this difference, see the next section, Reserved Instance IDs (p. 198).

- Your entire purchase of Reserved Instances within a region is completely within one 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 will be different from the new Reserved Instance IDs that will actually be created at the completion of the purchase. These IDs are returned when you use the describe command. For an explanation of this difference, see the next section, Reserved Instance IDs (p. 198).
- Your purchase of Reserved Instances within a region crosses from a lower discount tier to a higher discount tier. As with the previous scenario, 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 first or lower discounted rate, and an amount at the higher discounted rate. It is important to understand that the Reserved Instance IDs you get back will be different from the new Reserved Instance IDs that will actually be created at the completion of the purchase. For an explanation of this difference, see the next section, Reserved Instance IDs (p. 198).

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 AWS CLI, here's an example output showing the `aws ec2 describe-reserved-instances` command not recognizing the Reserved Instance ID `1ba8e2e3-edf1-43c3-b587-7742example`, which was returned by `aws ec2 purchase-reserved-instances-offering`.

```
{
    "ReservedInstances": []
}
```

In this example, the Reserved Instance ID `1ba8e2e3-edf1-43c3-b587-7742example` generated by the purchase command is an intermediate ID that is used while the purchase is being processed. It was not recognized by the Describe command. After the purchase is complete, you will get two new Reserved Instance IDs—an ID for the Reserved Instances purchased at the undiscounted price, and another ID for the Reserved Instances purchased at the discounted (tier 1) rate—that you can use and will be recognized by the Describe command.

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. For information about the Reserved Instances pricing tiers, see What Are the Reserved Instance Pricing Tiers? (p. 193). For information about Reserved Instances pricing, see Amazon EC2 Reserved Instance Pricing.

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 can modify them as long as requirements for modifications are met and capacity exists. For information about modifying Re-

served Instances, see Modifying Your Reserved Instances (p. 218). (You also 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 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 AWS Billing and Cost Management User Guide.

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 Reserved Instance Pricing Tiers (p. 193).

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. 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. 254).

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 Reserved Instance Pricing Tiers (p. 193)

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. 185).

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. 26).

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. 205).

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. 229).
- **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. 231).
- **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. 230).

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. 228).

What Do You Want to Do Next?

- Learn about:
  - Getting Started with Reserved Instances (p. 186)
  - Understanding the Pricing Benefit of Reserved Instances (p. 198)
  - Understanding Reserved Instance Pricing Tiers (p. 193)
- Start:
  - Obtaining Information About Your Reserved Instances (p. 213)
  - Purchasing Reserved Instances (p. 205)
  - Registering as a Seller (p. 229)
  - Listing Your Reserved Instance (p. 234)

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.

Note
To purchase and modify Reserved Instances, ensure that your account has the appropriate permissions, such as the ability to describe Availability Zones. For information, see the IAM Best Practices and the Permissions and Policies sections in the *Using IAM* guide.

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 modify your Reserved Instances and you can sell them if your needs change. For information about modifying your Reserved Instances, see Modifying Your Reserved Instances (p. 218). For information about selling your Reserved Instances in the Reserved Instance Marketplace, see Selling in the Reserved Instance Marketplace (p. 228).

This buyer's guide contains the following sections:

- How Buying Works (p. 203)—Provides an overview of what you need to get started buying in the Reserved Instance Marketplace.
- Becoming a Buyer (p. 204)—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. 205)—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. 211)—Helps you understand your bill.

For general information about the Reserved Instance Marketplace, see Reserved Instance Marketplace (p. 200). For information about selling Reserved Instances in the Reserved Instance Marketplace, see Selling in the Reserved Instance Marketplace (p. 228). For basic information about Reserved Instances, see Reserved Instances (p. 185).

For product pricing information, see the following pages:

- Amazon EC2 Reserved Instance Pricing
- Amazon EC2 On-Demand Instances Pricing
- AWS Service Pricing Overview

How Buying 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 pur-

chase 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 Reserved Instance Pricing Tiers (p. 193).

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 you buy Reserved Instances and 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. 204)
- Purchasing Reserved Instances (p. 205)

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. 26). If you are new to Reserved Instances, see Getting Started with Reserved Instances (p. 186).

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. 205).
- To understand your invoice, see Reading Your Statement (Invoice) (p. 211).
- To sell your unused Reserved Instances, see Selling in the Reserved Instance Marketplace (p. 228).

For information about Reserved Instances, see Reserved Instances (p. 185).

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 Amazon EC2 command line interface (CLI) tools, or the Amazon 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**.
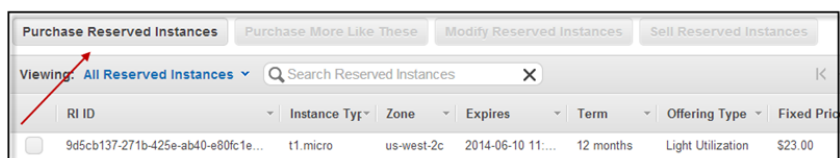
4. In the **Purchase Reserved Instances** page, specify the details of the Reserved Instances you want to purchase (use familiar filters like **Platform**, **Instance Type**, **Availability Zone**, **Term**, and **Tenancy**), and click **Search**.

   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.

6. 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.

7. 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.
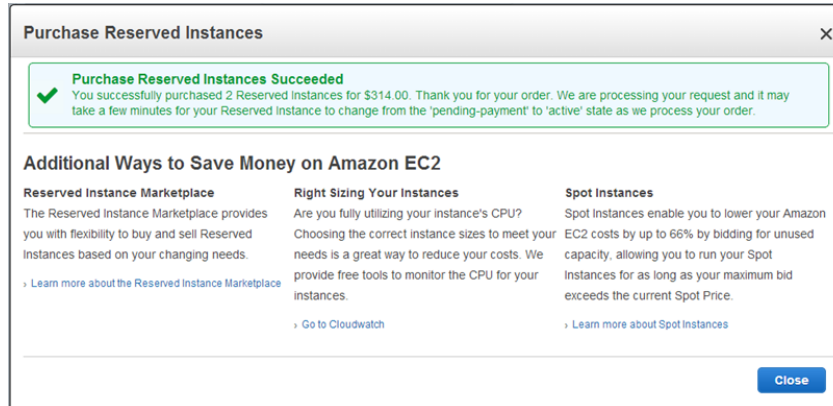
Your purchase is complete.

8. To verify your order, go to the **Reserved Instances** page in the EC2 console.
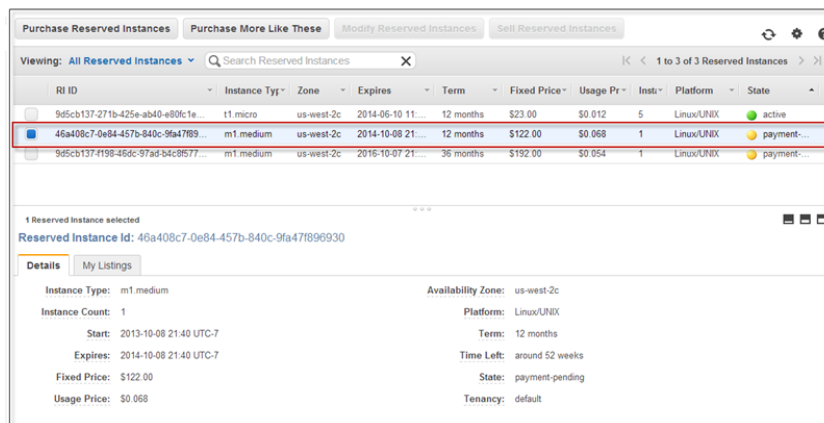
The **Reserved Instances** page displays a list of Reserved Instances that belong to your account, including the new Reserved Instance that you just purchased.

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-e4c11example 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-in-
> stances-offerings in the *Amazon EC2 Command Line Reference*.

2. From the list of available Reserved Instances, purchase the Reserved Instances that meet your re-
   quirements. 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.amazon
aws.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-a6400ex
ample</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 Reserved Instance you want and specify a limit price.

```
https://ec2.amazonaws.com/?Action=PurchaseReservedInstancesOffering
&ReservedInstancesOfferingId=d0280f9e-afc1-47f3-9899-c3a2cexample
&InstanceCount=1
&LimitPrice.Amount=200
&AUTHPARAMS
```

Following is an example response.

```
<PurchaseReservedInstancesOfferingResponse xmlns="http://ec2.amazon
aws.com/doc/2012-08-15/">
    <requestId>59dbff89-35bd-4eac-99ed-be587example</requestId>
    <reservedInstancesId>e5a2ff3b-7d14-494f-90af-0b5d0example</reservedIn
```

```
stancesId>
</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</re
servedInstancesId>
            <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.
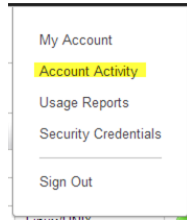
The **Account Activity** page will show all charges against your account—such as upfront and one-time fees and recurring charges. You can get both a summary of all your charges and a detailed list of your charges.

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.

You can view the charges online, and you can also download a PDF rendering of the charge information.

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.

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. 253).

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.
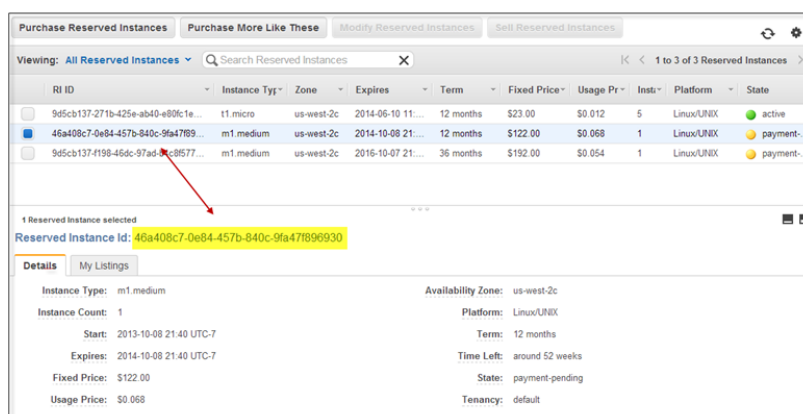
4. If you are a seller in the Reserved Instance Marketplace and you want information about your Reserved Instance listing, click the **My Listings** tab. You will see details about the Reserved Instance listing you selected.

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</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>
```

Modifying Your Reserved Instances

When your computing needs change, you can modify your Reserved Instances and continue to benefit from your capacity reservation. You can move your Reserved Instances between Availability Zones within the same region, and you can change the network platform between EC2-Classic and EC2-VPC. Availability Zone and network platform modifications are supported for all product platform types.

In addition, starting with Amazon EC2 API version 2013-10-01, you can change the instance type of your reservation to a larger or smaller instance type in the same family (e.g., the M1 instance family), if capacity is available, and if you are modifying a Reserved Instance associated with the Linux/UNIX product platform. Instance type modifications are not supported for other platform types.

Modification does not change the remaining term of your Reserved Instances. Their end dates remain the same. There is no fee for modifying your Reserved Instances, and you do not receive any new bills or invoices.

Instances covered by Reserved Instances continue to run even if you modify your reservation. However, after modification, the pricing benefit of the Reserved Instances starts applying to instances that match the new parameters of your Reserved Instances. You are charged at the On-Demand rate for the EC2 instances no longer receiving the benefits of the modified Reserved Instances, unless your account has other applicable reservations, in which case you will be charged at the rate of these other Reserved Instances.

Modification does not affect how you use, purchase, or sell Reserved Instances. When you purchase Reserved Instances, you still need to specify the product platform, instance type, network platform, tenancy, term length, offering type, and Availability Zone.

> **Note**
> To purchase and modify Reserved Instances, ensure that your account has the appropriate permissions, such as the ability to describe Availability Zones. For information, see the IAM Best Practices and the Permissions and Policies sections in the *Using IAM* guide.

You can view the status of your request in the AWS Management Console, or when you call the `DescribeReservedInstancesModifications` API action or `ec2-describe-reserved-instances-modifications` CLI command.

This section discusses the modification process:

- Understanding the Modification Process (p. 218)—What happens when I submit a modification request?
- Determining the Status of Your Modification (p. 220)—How do I track my modification requests?
- Requirements for Modification (p. 221)—Which Reserved Instances can I modify?
- Changing the Instance Type of Your Reservations (p. 222)—When can I change the instance size of my Reserved Instances?
- Submitting Modification Requests (p. 224)—How do I modify my Reserved Instances?

Understanding the Modification Process

You can modify your Reserved Instances in one or more of the following ways:

- Move them between Availability Zones within the same region.

If you modify the Availability Zone of your Reserved Instances, the capacity reservation and pricing benefit stop applying to instance usage in the original zone, and start applying to usage in the new Availability Zone.

- Change the network platform of the Reserved Instances between EC2-VPC and EC2-Classic.

  If you modify the network platform of your Reserved Instances, the capacity reservation stops applying to instance usage with the original network platform, and starts applying to usage with the new network platform. However, the pricing benefit continues to apply to both EC2-Classic and EC2-VPC instance usage matching the remaining Reserved Instances parameters.

- Upgrade or downgrade the instance type of your Reserved Instances within the same instance family (e.g., the M1 instance family).

  If you modify the instance type of your Reserved Instances, you must ensure that the instance family of your reservation has a larger or smaller instance type available and that you have enough applicable Reserved Instances to make the change.

You can modify your whole reservation or a subset of your reservation. When you modify a subset of your reservation, Amazon EC2 splits your original Reserved Instances into two or more new Reserved Instances. For example, if you have Reserved Instances for 10 instances in us-east-1a, and decide to move 5 instances to us-east-1b, the modification request results in two new Reserved Instances—one for 5 instances in us-east-1a (the original zone), and the other for 5 instances in us-east-1b.

Amazon EC2 fulfills your modification request as soon as possible, depending on available capacity. Until your modification request completes, the capacity reservation and pricing benefit associated with your Reserved Instances continue to be based on the original parameters of your reservation.

> **Note**
> You cannot cancel or change a pending modification request after you submit it. While your modification is being processed, the status of the Reserved Instances that you're modifying is *active (pending modification)*. After the modification has completed successfully, you can submit another modification request to roll back any changes you made.

If your Reserved Instances modification request succeeds:

- The modified reservation becomes effective immediately and the pricing benefit of the Reserved Instances is applied to the new instances beginning at the hour of the modification request. For example, if you successfully modify your Reserved Instances at 9:15PM, the pricing benefit transfers to your new instance at 9:00PM. (You can get the *effective date* of the modified Reserved Instances by using the `DescribeReservedInstances` API action or the `ec2-describe-reserved-instances` CLI command.)

- The end date of the modified Reserved Instances is the same as the original end date of the reservation. If you successfully modify a three-year reservation that had 16 months left in its term, the resulting modified reservation is a 16-month Reserved Instance with the same end date as the original Reserved Instances.

  For example, in the following table, Modification #1 shows RI af9f760... was modified successfully and retired on *2013-08-30 16:00 UTC-7* and a new RI 46a408c... was created as a result of the modification on the same date-time that RI af9f760... was retired. Modification #2 shows RI 46a408c... was modified successfully and retired on *2013-09-03 14:00 UTC-7* and a new RI b847fa9... was created as a result of the modification on the same date-time that RI 46a408c... was retired.

- The original reservation is retired. Its end date is the start date of the new reservation, and the end date of the new reservation is the same as the end date of the original Reserved Instance when it was active.
- The modified Reserved Instances shows $0 fixed price and not the fixed price of the original Reserved Instances.

> **Note**
> The fixed price of the modified reservation does not affect the discount tier calculations applied to your account, which are based on the fixed price of the original reservation.

If your modification request fails:

- Your Reserved Instances maintain the original properties that they had prior to your request.
- Your Reserved Instances are available for another modification request.

You can determine the status of your request by looking at the *state* of the Reserved Instances that you are modifying. For information, see Determining the Status of Your Modification (p. 220).

Determining the Status of Your Modification

The state of your modification request is displayed in the **State** field in the AWS Management Console. You can also use the `DescribeReservedInstancesModifications` API action to get detailed information about your modification request: The `state` returned shows your request as `in-progress`, `fulfilled`, or `failed`.

You can only modify your Reserved Instances if they are *active*. You are not able to modify Reserved Instances that are in any other state. You also cannot modify your Reserved Instances if they are listed in the Reserved Instance Marketplace. For more information, see Requirements for Modification (p. 221). If your Reserved Instances are not in the active state or cannot be modified, the **Modify Reserved Instances** button in the AWS Management Console is not enabled. If you use the API to modify Reserved Instances that are not active, you will get an error.

In cases when you select Reserved Instances whose Availability Zone and network platform can be modified, but whose instance types cannot be changed, the **Modify Reserved Instances** button is enabled and you can proceed to the **Modify Reserved Instances** page. However, you do not have the option to modify the instance type. For more information, see Changing the Instance Type of Your Reservations (p. 222).

Modification States

The modification request is being processed. While the modification request is being processed, the status of the Reserved Instances being modified show as *active (pending modification)*. The Reserved Instances are in this state only for a short period. The state reverts to *active* or becomes *retired*, depending

on the success of the modification. (If you use the `DescribeReservedInstancesModifications` API action, the status of your modification request should show *processing.*)

The modification succeeded. If the modification is successful, the Reserved Instances being modified are *retired*, and new Reserved Instances are created with the modification configuration that you requested. The status of these new Reserved Instances is *active*. (If you use the `DescribeReservedInstances-Modifications` API action, the status of your modification request should show *fulfilled.*)

> **Note**
> For the brief period that the new Reserved Instances are being activated, the original Reserved Instances show as *retired (pending modification).*

The modification failed. If the modification did not complete, the Reserved Instances being modified return to the *active* state. (If you use the `DescribeReservedInstancesModifications` API action, the status of your modification request should show *failed.*) For information about why some Reserved Instances cannot be modified, see Requirements for Modification (p. 221).

Requirements for Modification

Amazon EC2 processes your modification request if the following conditions are met:

- We have sufficient Reserved Instances capacity for your target configuration.
- The Reserved Instances you are modifying are active.
- The Reserved Instances are not pending another modification request.

  You may modify your Reserved Instances as frequently as you like. However, you cannot submit a modification request for Reserved Instances that are still pending a previous modification request; that is, if the previous modification request is in the *active (pending modification)* state. For information about modification states, see Determining the Status of Your Modification (p. 220).
- The Reserved Instances are not listed in the Reserved Instance Marketplace.

  To modify Reserved Instances that are listed in the Reserved Instance Marketplace, cancel the listing, modify the Reserved Instances, and then list them again.

  In addition, you cannot modify a Reserved Instance Marketplace offering before or at the same time that you purchase it. However, you can submit a modification request after you purchase a Marketplace offering. For more information, see Reserved Instance Marketplace (p. 200).
- Only the following attributes are being modified: Availability Zone in the same Region, instance type in the same instance family (e.g., the M1 instance family), and network platform.
- When combining Reserved Instances, the end dates of the Reserved Instances must have the same hour digits in 24-hour format, but they do not need to have the same minutes or seconds. This means you can combine Reserved Instances if they end on the same day at 13:01 and 13:59, but not if they end at 12:59 and 13:01.
- When modifying the instance type attribute
  - The Reserved Instances you are modifying must be for the Amazon Linux/UNIX product platform (currently the only product platform for which instance type modifications are allowed).
  - The new instance type of the Reserved Instances must be in the same instance family (e.g., the M1 instance family) as the original reservation.
  - If you are upgrading the instance type of your Reserved Instances, the normalized instance count for the target Reserved Instance configuration must equal the normalized instance count of the original reservation.

  If you select multiple Reserved Instances for modification and one or more of these Reserved Instances are for a product platform that does not allow instance type modification, the **Modify Reserved Instances** page will not show the option of changing the instance type for any of the selected Reserved Instances. For information about modifying instance types, see Changing the Instance Type of Your Reservations (p. 222).

- The attributes for the target Reserved Instance configuration must result in a unique combination of Availability Zone, instance type, and network platform.

For information about how to modify Reserved Instances, see Submitting Modification Requests (p. 224).

Changing the Instance Type of Your Reservations

Under certain conditions, you can adjust the instance type of your Reserved Instances. If you have capacity reservations for Amazon Linux/UNIX in instance families with multiple instance sizes, you can request to modify your Reserved Instances to include different instance types within the same family. Your request proceeds successfully if the capacity exists and the modification does not change the instance size footprint of your Reserved Instances.

For example, you can divide a reservation for one m1.large instance into four m1.small instances, or you can combine a reservation for four m1.small instances into one m1.large instance. In either case, the instance size footprint of the reservation does not change. On the other hand, you cannot change your reservation for two m1.small instances into one m1.large instance because the existing instance size footprint of your current reservation is smaller than the proposed reservation.

Instance size footprints are determined by the normalization factor of the instance type and the number of instances in the reservation. This section discusses the two ways you can change the instance type of your Reserved Instances, and how you use the instance type's normalization factor to figure out what instance type modifications you can make.

- Understanding the Instance Normalization Factor (p. 223)
- Upgrading Your Instance Type (p. 223)
- Downgrading Your Instance Type (p. 224)

Exceptions

The following instance type sizes cannot be modified because there are no other sizes in their families.

- t1.micro
- cc1.4xlarge
- cc2.8xlarge
- cg1.8xlarge
- cr1.8xlarge
- hi1.4xlarge
- hs1.8xlarge
- g2.2xlarge

In addition, instance type modifications are currently supported only for Linux/UNIX platform types. For more information, see Requirements for Modification (p. 221).

When you try to modify Reserved Instances that have instance types or product platforms that cannot be modified, you can proceed to the **Modify Reserved Instances** page, but the Amazon EC2 console limits the modification options to **Network** and **Availability Zone**.

Keep in mind that instance type modifications are allowed only if other Reserved Instances specification details match—such as region, utilization type, tenancy, product, end date and hour—and capacity is available. In addition, such modifications do not change the discounted hourly usage fee that you are billed, or the end date of your reservation.

Understanding the Instance Normalization Factor

Each Reserved Instance has an instance size footprint. When you modify the instance type of Reserved Instances, the footprint is maintained even if the instance type is downsized or upsized. A modification request is not processed if the footprint of the target configuration does not match the size of the original configuration.

The size of an instance type's footprint can be calculated by using its normalization factor, which is based on the type's size within the instance family (e.g., the M1 instance family). Normalization factors are only meaningful within the same instance family; instance types cannot be modified from one family to another.

The following table illustrates the normalization factor that applies within an instance family. For example, an m1.small instance has a normalization factor of 1, an m1.medium instance has a factor of 2, and an m1.large instance has a factor of 4.

Instance Size	Normalization Factor
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
4xlarge	32
8xlarge	64

Each Reserved Instance has a total number of *normalized instance units*, which is equal to the instance count multiplied by the normalization factor of the instance type. For example, an m1.medium has a normalization factor of 2 so a Reserved Instance for four m1.medium instances is worth eight normalized instance units. You arrive at the value this way:

4 [count] x 2 [normalization factor]

You can allocate your Reserved Instances into different instance sizes across the same instance family (e.g., the M1 instance family) as long as the total normalized instance units of your Reserved Instances remain the same. If you have Reserved Instances for four m1.medium instances, you can turn it into a reservation for eight m1.small instances (m1.small instances have a normalization factor of 1, thus 8 x 1). The resulting Reserved Instances would have the same normalized instance units or instance size footprint.

Upgrading Your Instance Type

You can consolidate a reservation of multiple smaller instance types that belong to one instance family (e.g., the M1 instance family) into a reservation for a larger instance type in the same family. To upgrade or *upsize* your reservation, you must have enough of the smaller instance types in the same reservation to consolidate into larger instance types, and your reservation's overall instance size footprint does not change.

For example, you can convert four m1.small instances, which is equivalent to four normalized instance units (4 x 1), into one m1.large instance, which is also equivalent to four normalized units (1 x 4). However, you cannot convert a reservation for a single m1.small instance (1 x 1) into a reservation for an m1.large instance (1 x 4). The two reservations are not equal.

Instance families that have multiple instance sizes, and are not listed in the Exceptions (p. 222) list, can be changed or upgraded to larger instance type sizes.

Downgrading Your Instance Type

You can change Reserved Instances for a large instance type to several Reserved Instances of smaller instance types of the same family (e.g., the M1 instance family). When you downgrade or *downsize* your Reserved Instances, you are actually dividing a large reservation into multiple smaller reservations. Just as with upsizing instance types, you can downsize successfully only if your reservation's overall instance size footprint does not change.

For example, a reservation for two m1.large instances equals eight normalized instance units (2 x 4). Assuming there is capacity, they can be converted to a Reserved Instance for four m1.medium instances, which is also equivalent to eight normalized instance units (4 x 2).

Instance families that have multiple instance sizes, and are not listed in the Exceptions (p. 222) list, can be changed or downgraded to smaller instance type sizes.

For information about Amazon EC2 instance types, see Instance Types (p. 103). For information about Reserved Instances modification requests, see Submitting Modification Requests (p. 224).

Submitting Modification Requests

You can use the AWS Management Console to modify your Reserved Instances in the following ways:

- Move all or some of your capacity reservation to another Availability Zone in the same region.
- Change the network platform of all or some of your Reserved Instances from EC2-Classic to EC2-VPC, or vice versa.
- Upgrade or downgrade all or some of the instance types of your Reserved Instances within the same instance family (e.g., the M1 instance family) as long as there is capacity and the change maintains the footprint of your capacity reservation.

You can also complete these modification tasks programmatically by using the AWS CLI (modify-reserved-instances), Amazon EC2 CLI (ec2-modify-reserved-instances), Amazon EC2 API (ModifyReservedInstances), and the AWS SDK for Java.

For information about programming tools you can use to modify Reserved Instances, see Tools for Working with Reserved Instances (p. 189).

Using the Modifying Reserved Instances Page

In this section, you will learn how to use the **Modify Reserved Instances** page in the Amazon EC2 Console to make modifications. We will also discuss the following:

- Count and Units Values (p. 224)
- Messages About Your Modification Requests (p. 227)

Count and Units Values

The **Modify Reserved Instances** page in the EC2 console keeps track of the total count and units of the Reserved Instances that you have allocated for the target (or new) configurations versus the total number of Reserved Instances available for you to modify. **Count** is the literal number of instances for the current instance type. **Units** represent the total instance size of your reservation relative to its instance family (e.g., the M1 instance family). The value of the units equals the count multiplied by the normalization factor of the instance type. For information about the instance normalization factor, see Understanding the Instance Normalization Factor (p. 223).

Each target configuration row for your Reserved Instances has a count and units value. The allocated total is displayed in red if you have specified either more or fewer Reserved Instances than are available for modification. You cannot click **Continue** if the total units you have specified for your target configurations do not equal the total number of units of Reserved Instances available to modify. The units displayed will change to green after you have specified changes for all the Reserved Instances that were available for modification.

To modify your Reserved Instances

1. On the **Reserved Instances** page, select one or more Reserved Instances to modify, and click **Modify Reserved Instances**.
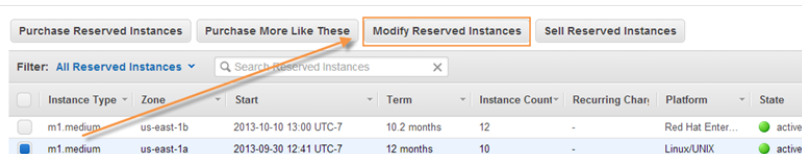
   The **Modify Reserved Instances** page opens with the Reserved Instances you selected and their current configuration.

2. Decide how to modify your Reserved Instances. For this example, you want to modify your Reserved Instances so that 2 are not changed, 2 are moved to another Availability Zone, 2 are converted from EC2-Classic to EC2-VPC, and 2 are upgraded from m1.medium to m1.large instance type. Keep in mind that to upgrade, you will need 2 m1.medium instances for every 1 m1.large instance.

   The following table illustrates the changes that you want to implement:

Modification	Network	Availability Zone	Instance Type	Pre-Change Count	Post-Change Count	Units
No change (Keep some of the original RIs)	EC2-Classic	us-east-1a	m1.medium	2	2	4
Change Availability Zone	EC2-Classic	*us-east-1b*	m1.medium	2	2	4
Change network platform	*EC2-VPC*	us-east-1a	m1.medium	2	2	4

Modification	Network	Availability Zone	Instance Type	Pre-Change Count	Post-Change Count	Units
Change instance type	EC2-Classic	us-east-1a	*m1.large*	4	2	8
Total				10	8	20

3. To keep the original configuration (no change) for two Reserved Instances, type *2* for **Count**. **Network**, **Availability Zone**, and **Instance Type** stay the same. **Units** automatically updates to *4/20*.

   Use the **Add** button to add a row so that each change you want to make to your Reserved Instances is displayed in a separate row.

4. To change the Availability Zone, click **Add** and select *us-east-1b* in the **Availability Zone** list. For **Count**, type *2*. **Network** and **Instance Type** stay the same. **Units** automatically updates to *8/20*.

5. To change the network platform, first click **Add**. Then select *EC2-VPC* in the **Network** list. For **Count**, type *2*. The **Availability Zone** and **Instance Type** stay the same. **Units** automatically updates to *12/20*.

6. To change the instance type, click **Add** and select *m1.large* in the **Instance Type** list, and, for **Count**, type *2*. The **Availability Zone** and **Network** stay the same. **Units** automatically updates to *20/20*.

   If you want to change an instance type, be sure that it can be converted to a larger or smaller set of instance types. For more information about changing instance types, see Changing the Instance Type of Your Reservations (p. 222).

In our example, you are upgrading from an m1.medium to an m1.large instance type, which has a larger footprint. The normalization factor for m1.large instance types is 4; to get one m1.large instance type you'll need 4 m1.small or 2 m1.medium instance types. Therefore, the remaining 4 m1.medium instance types, which have a normalized value of 8, can be upgraded to 2 m1.large instance types, which also have a normalized value of 8. **Units** will be *20/20* in green, which means that you have specified changes for all the Reserved Instances that were available for modification in this exercise and you can proceed with your modification request.

For more information about the instance normalization factor, see Understanding the Instance Normalization Factor (p. 223).

7. If you want to delete a configuration that you specified, such as the instance upgrade, click **X** for that row.

    Note
    If the **Modify Reserved Instances** page contains only one row for configuration changes, you cannot delete that row.

8. To confirm your modification choices when you finish specifying your target configurations, click **Submit Modifications**. Amazon EC2 checks your modification request. If the configuration settings you requested were unique, the instance count that you specified matches the total number of Reserved Instances available for you to modify, you will be informed that Amazon EC2 is processing your request.

    Note
    At this point, Amazon EC2 has only determined that the parameters of your modification request are valid, and Amazon EC2 can now process the request. Only during the processing of the modification request, after you get this screen, is the capacity check made. Your modification request can still fail, but at a later point, due to capacity not available.

Messages About Your Modification Requests

In some situations, you might get a message indicating incomplete or failed modification requests instead of a confirmation. Use the information in such messages as a starting point for resubmitting another modification request.

*   Not all selected Reserved Instances can be processed for modification.

    In the following message, Amazon EC2 identifies and lists the Reserved Instances that cannot be modified. If you receive a message like this, go to the **Reserved Instances** page in the EC2 console and check the information details about these capacity reservations.

*   Error in processing your modification request.

    You submitted one or more Reserved Instances for modification and none of your requests can be processed. Depending on the number of Reserved Instances you are modifying, you can get different versions of the message.

> ⚠ **Error**
> The error(s) below occurred while we were processing your Modify Reserved Instances request. If you cannot resolve them, please contact AWS Customer Support.
>
> Invalid value for 'targetReservedInstancesConfigurations'. Configuration LeaseConfiguration(availabilityZone=us-east-1a, targetPlatform=EC2-Classic, instanceCount=1) is a duplicate.

> ⚠ **Error**
> The error(s) below occurred while we were processing your Modify Reserved Instances request. If you cannot resolve them, please contact AWS Customer Support.
>
> Invalid value for 'targetReservedInstancesConfigurations'. Configuration LeaseConfiguration(availabilityZone=us-east-1a, targetPlatform=EC2-VPC, instanceCount=1) is a duplicate.
>
> Invalid value for 'targetReservedInstancesConfigurations'. Configuration LeaseConfiguration(availabilityZone=us-east-1a, targetPlatform=EC2-VPC, instanceCount=1) is a duplicate.
>
> Invalid value for 'targetReservedInstancesConfigurations'. Configuration LeaseConfiguration(availabilityZone=us-east-1a, targetPlatform=EC2-Classic, instanceCount=1) is a duplicate.

In these messages, Amazon EC2 displays the reasons why your request cannot be processed. For example, you might have specified the same target configuration—a combination of Availability Zone and platform—for one or more subsets of the Reserved Instances you are modifying. Try submitting these modification requests again, but ensure that instance details of the Reserved Instances match, and that the target configurations for all subsets of the Reserved Instances being modified are unique.

What do you want to do next?

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. 254).)

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. 229)—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. 233)—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. 252)—Find out when your Reserved Instance is sold, and how you get paid.
- Quick Start: Selling in the Reserved Instance Marketplace Video (p. 202)—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. 202). For basic information about Reserved Instances, see Reserved Instances (p. 185).

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. 229).
- 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 Reserved Instance Pricing Tiers (p. 193).

For a checklist that summarizes requirements for working with Reserved Instances and the Reserved Instance Marketplace, see Requirements Checklist for Reserved Instances (p. 254).

Registering as a Seller

Topics
- Your Bank (p. 230)
- Tax Information (p. 231)
- Seller Registration Confirmation (p. 232)
- Sharing Information with the Buyer (p. 232)
- Next Steps (p. 232)

To be able to sell in the Reserved Instance Marketplace, your first task is to register as a seller.
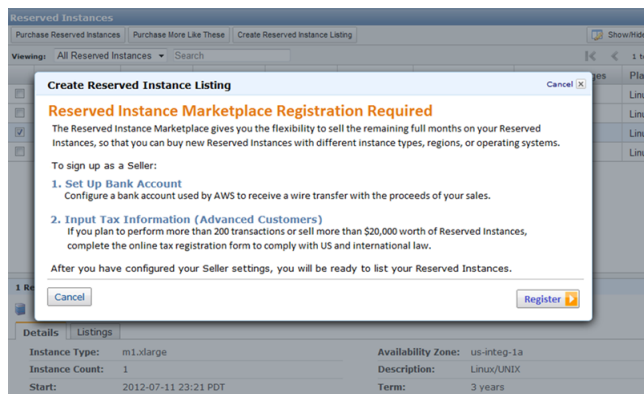
If you haven't created an AWS account yet, you need to do this first before you register for Reserved Instance Marketplace. Complete the instructions described in Getting Started with Amazon EC2 Linux Instances (p. 26), which provides information about creating your Amazon EC2 account and credentials.

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

  > **Note**
  > 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.

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 ."
- 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. 234).

2. Listing your Reserved Instances.

   Include your Reserved Instances in the Reserved Instance Marketplace listings. For more information, see Listing Your Reserved Instance (p. 234).

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. 213).

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. 243).

For information about selling your Reserved Instances, see Selling Your Reserved Instances (p. 233).

Selling Your Reserved Instances

Topics

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. 233).

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. 254).

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. 189).

- The ec2-describe-reserved-instances CLI command.
- The DescribeReservedInstances API action.

For more information about Reserved Instances, see Reserved Instances (p. 185).

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. 191).

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**.
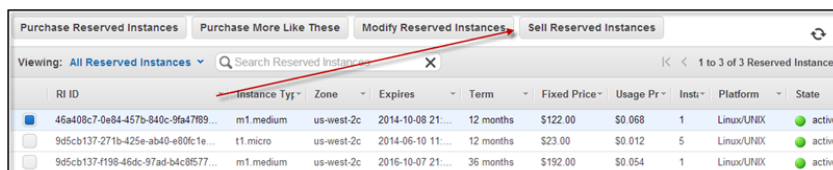
> **Note**
> If you have not completed the Reserved Instance Marketplace seller registration process, you will be prompted to complete this process now. For information about the seller registration process, see Registering as a Seller (p. 229).

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.

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 **Reset**. Click **Continue** when you are finished configuring your listing.

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.
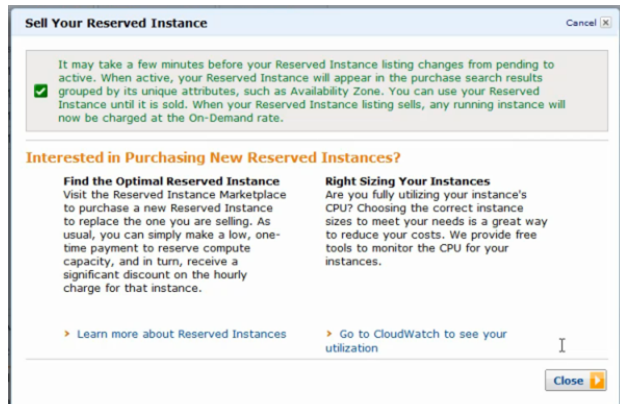
7. To view the details of your Reserved Instance listing, on the **Reserved Instances** page, select the Reserved Instance you want to view, and click the **My Listings** tab.
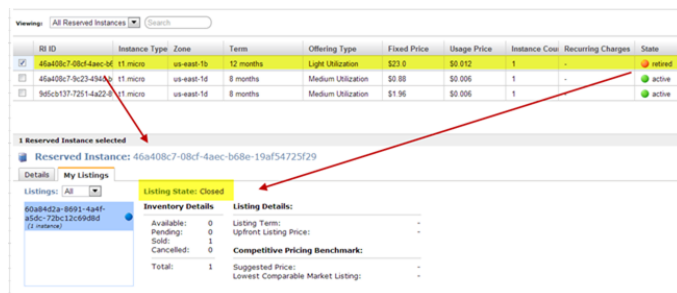
Amazon EC2 CLI

To list a Reserved Instance in the Reserved Instance Marketplace

1. 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 1ba8e2e3-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.

2. 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</reserved
InstancesId>
        <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>
```

Canceling 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. 243), the Amazon EC2 CLI (p. 244), or the Amazon EC2 API (p. 246). 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.
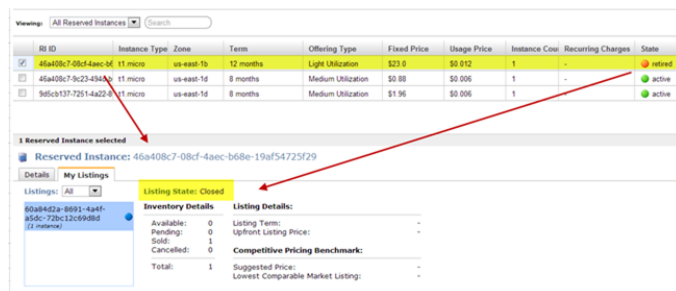
To cancel your listing

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**.

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 .

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 can
celled 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. 234).

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>
                    <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. 234).

After Your Reserved Instance Is Sold

Topics
- Reserved Instance Listing States (p. 253)
- Lifecycle of a Listing (p. 253)
- Getting Paid (p. 254)
- Notifications (p. 254)

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. 254).

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. 213).

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.

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	m1.large	Medium Utilization	2 months	us-east-1a	$24.44	2	$39.11

Recent Sales

Platform	Instance Type	Offering Type	Term	Availability Zone	Price	Count	Total Earnings
Linux/UNIX (Amazon VPC)	m1.large	Heavy Utilization	35 months	us-east-1a	$1,480.50	1	$1,317.64
Linux/UNIX (Amazon VPC)	m1.large	Medium Utilization	35 months	us-east-1a	$3,850.00	1	$3,426.50

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.

Note
To purchase and modify Reserved Instances, ensure that your account has the appropriate permissions, such as the ability to describe Availability Zones. For information, see the IAM Best Practices and the Permissions and Policies sections in the *Using IAM* guide.

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. 26), 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. 191).
- **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 Reserved Instance Pricing Tiers (p. 193).
- **Cancellation 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, see Selling Your Reserved Instances (p. 233).

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. 233).
- **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. 230).
- **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. 231).
- **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. 243). You also can change your Reserved Instances before listing them. For information, see Modifying Your Reserved Instances (p. 218).
- **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, see Reserved Instance Marketplace (p. 200).

- **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 in the Reserved Instance Marketplace.

Instance Metadata and User Data

Instance metadata is data about your instance that you can use to configure or manage the running instance. Instance metadata is divided into categories. For more information, see Instance Metadata Categories (p. 262).

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. 266).

You can also access the *user data* that you supplied when launching your instance. For example, you can specify parameters for configuring your instance, or attach a simple script. You can also use this data to build more generic AMIs that can be modified by configuration files supplied at launch time. For example, if you run web servers for various small businesses, they can all use the same AMI and retrieve their content from the Amazon S3 bucket you specify in the user data at launch. To add a new customer at any time, simply create a bucket for the customer, add their content, and launch your AMI. If you launch more than one instance at the same time, the user data is available to all instances in that reservation.

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

> **Important**
> Although you can only access instance metadata and user data from within the instance itself, the data is not protected by cryptographic methods. Anyone who can access the instance can view its metadata. Therefore, you should take suitable precautions to protect sensitive data (such as long-lived encryption keys). You should not store sensitive data, such as passwords, as user data.

For more information about adding user data when you launch an instance, see Launching an Instance (p. 290). You can add or modify user data on Amazon EBS-backed instances when they're stopped. For more information about adding user data to a stopped instance, see Modifying a Stopped Instance (p. 311).

When you are adding user data, take note of the following:

- User data is treated as opaque data: what you give is what you get back. It is up to the instance to be able to interpret it.
- User data is limited to 16 KB. This limit applies to the data in raw form, not base64-encoded form.
- User data must be base64-encoded before being submitted to the API. The API command line tools perform the base64 encoding for you. The data is decoded before being presented to the instance. For more information about base64 encodings, go to http://tools.ietf.org/html/rfc4648.

Topics

- Retrieving Instance Metadata (p. 257)
- Retrieving User Data (p. 259)
- Retrieving Dynamic Data (p. 260)

Retrieving Instance Metadata

To view all categories of instance metadata from within a running instance, use the following URI:

```
http://169.254.169.254/latest/meta-data/
```

Note that you are not billed for HTTP requests used to retrieve instance metadata and user data.

You can use a tool such as cURL, or if your instance supports it, the GET command; for example:

```
$ curl http://169.254.169.254/latest/meta-data/
```

```
$ 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

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

A request for a general metadata resource (the URI ends with a /) returns a list of available resources, or a `404 - Not Found` HTTP error code if there is no such resource. The list items are on separate lines, terminated by line feeds (ASCII 10).

Examples of Retrieving Instance Metadata

This example gets the available versions of the instance metadata. These versions do not necessarily correlate with an Amazon EC2 API version. The earlier versions are available to you in case you have scripts that rely on the structure and information present in a previous version.

```
$ curl http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
latest
```

This example gets the top-level metadata items. Some items are only available for instances in a VPC. For more information about each of these items, see Instance Metadata Categories (p. 262).

```
$ curl http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
instance-action
instance-id
instance-type
kernel-id
local-hostname
local-ipv4
mac
network/
placement/
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
```

These examples get the value of some of the metadata items from the preceding example.

```
$ curl http://169.254.169.254/latest/meta-data/ami-id
ami-2bb65342
```

```
$ curl http://169.254.169.254/latest/meta-data/reservation-id
r-fea54097
```

```
$ curl http://169.254.169.254/latest/meta-data/hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

This example gets the list of available public keys.

```
$ curl 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.

```
$ curl http://169.254.169.254/latest/meta-data/public-keys/0
openssh-key
```

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

```
$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
```

```
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
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.

```
$ curl http://169.254.169.254/latest/meta-data/network/inter
faces/macs/02:29:96:8f:6a:2d/
device-number
local-hostname
local-ipv4s
mac
owner-id
public-hostname
public-ipv4s
```

This example gets the subnet ID for an instance launched into a VPC.

```
$ curl http://169.254.169.254/latest/meta-data/network/inter
faces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

Retrieving User Data

To retrieve user data, use the following URI:

```
http://169.254.169.254/latest/user-data
```

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

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

```
$ curl http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

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

```
$ curl http://169.254.169.254/latest/user-data
[general]
instances: 4

[instance-0]
s3-bucket: <user_name>

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

Retrieving Dynamic Data

To retrieve dynamic data from within a running instance, use the following URI:

```
http://169.254.169.254/latest/dynamic/
```

This example shows how to retrieve the high-level instance identity categories:

```
$ curl http://169.254.169.254//latest/dynamic/instance-identity/
pkcs7
signature
document
```

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:

```
$ ec2-run-instances ami-2bb65342 -n 4 -d "replicate-every=1min | replicate-
every=5min | replicate-every=10min"

RESERVATION     r-fea54097      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: r-fea54097
- Public keys: none

- Security group name: default
- Instance type: m1.small

However, each instance has certain unique metadata.

Instance 1

Metadata	Value
instance-id	i-10a64379
ami-launch-index	0
public-hostname	ec2-203-0-113-25.compute-1.amazonaws.com
public-ipv4	67.202.51.223
local-hostname	ip-10-251-50-12.ec2.internal
local-ipv4	10.251.50.35

Instance 2

Metadata	Value
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

Metadata	Value
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

Metadata	Value
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
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:

```
$ curl http://169.254.169.254/latest/meta-data/ami-launch-index
2
```

2. She saves the ami-launch-index as a variable:

```
$ ami_launch_index=`curl http://169.254.169.254/latest/meta-data/ami-launch-
index`
```

3. She saves the user data as a variable:

```
$ user_data=`curl 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:

```
$ 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

Data	Description	Version Introduced
`ami-launch-index`	If you started more than one instance at the same time, this value indicates the order in which the instance was launched. The value of the first instance launched is 0.	1.0
`ami-manifest-path`	The path to the AMI's manifest file in Amazon S3. If you used an Amazon EBS-backed AMI to launch the instance, the returned result is `unknown`.	1.0
`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 `ancestor-amis` 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` `N`	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 *N* indicates the index of the Amazon EBS volume (such as `ebs1` or `ebs2`).	2007-12-15
`block-device-mapping/eph emeral` `N`	The virtual devices associated with ephemeral devices, if any are present. The *N* 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 eth0 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.	2012-01-12

Data	Description	Version Introduced
`iam/security-credentials /role-name`	Where *role-name* 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.	2012-01-12
`instance-action`	Notifies the instance that it should reboot in preparation for bundling. Valid values: `none` | `shutdown` | `bundle-pending`.	2008-09-01
`instance-id`	The ID of this instance.	1.0
`instance-type`	The type of instance. For more information, see Instance Types (p. 103).	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/pu blic-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

Data	Description	Version Introduced
`network/interfaces/macs/` `mac/mac`	The instance's MAC address.	2011-01-01
`network/interfaces/macs/` `mac/owner-id`	The ID of the owner of the network interface. In multiple-interface environments, an interface can be attached by a third party, such as Elastic Load Balancing. Traffic on an interface is always billed to the interface owner.	2011-01-01
`network/interfaces/macs/` `mac/public-hostname`	The interface's public DNS. If the instance is in a VPC, this category is only returned if the `enableDnsHostnames` attribute is set to `true`. For more information, see Using DNS with Your VPC.	2011-01-01
`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 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 instances launched into a VPC. For more information on security groups in the EC2-VPC platform, see Security Groups for Your VPC.	2011-01-01
`network/interfaces/macs/` `mac/subnet-id`	The ID of the subnet in which the interface resides. Returned only for instances launched into a VPC.	2011-01-01
`network/interfaces/macs/` `mac/subnet-ipv4-cidr-block`	The CIDR block of the subnet in which the interface resides. Returned only for instances launched into a VPC.	2011-01-01
`network/interfaces/macs/` `mac/vpc-id`	The ID of the VPC in which the interface resides. Returned only for instances launched into a VPC.	2011-01-01
`network/interfaces/macs/` `mac/vpc-ipv4-cidr-block`	The CIDR block of the VPC in which the interface resides. Returned only for instances launched into a VPC.	2011-01-01
`placement/availability-z` `one`	The Availability Zone in which the instance launched.	2008-02-01
`product-codes`	Product codes associated with the instance, if any.	2007-03-01

Data	Description	Version Introduced
`public-hostname`	The instance's public DNS. If the instance is in a VPC, this category is only returned if the `enableDnsHostnames` attribute is set to `true`. For more information, see Using DNS with Your VPC.	2007-01-19
`public-ipv4`	The public IP address. If an Elastic IP address is associated with the instance, the value returned is the Elastic IP address.	2007-01-19
`public-keys/0/openssh-key`	Public key. Only available if supplied at instance launch time.	1.0
`ramdisk-id`	The ID of the RAM disk specified at launch time, if applicable.	2007-10-10
`reservation-id`	ID of the reservation.	1.0
`security-groups`	The names of the security groups applied to the instance. **Note** Only instances launched into a VPC can change security groups after launch. These changes will be reflected here and in network/interfaces/macs/*mac*/security-groups.	1.0
`services/domain`	The domain for AWS resources for the region; for example, `amazonaws.com` for us-east-1.	2014-02-25

Dynamic Data Categories

The following table lists the categories of dynamic data.

Data	Description	Version introduced
`fws/instance-monitoring`	Value showing whether the customer has enabled detailed one-minute monitoring in CloudWatch. Valid values: `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

Importing and Exporting Instances

You can use the Amazon Web Services (AWS) VM Import/Export tools to import virtual machine (VM) images from your local environment into AWS and convert them into ready-to-use Amazon EC2 instances. Later, you can export the VM images back to your local environment. VM Import/Export allows you to leverage your existing investments in the virtual machines that you have built to meet your IT security, configuration management, and compliance requirements by bringing those VMs into Amazon Elastic Compute Cloud (Amazon EC2) as ready-to-use instances. VM Import/Export is compatible with Citrix Xen, Microsoft Hyper-V, or VMware vSphere virtualization environments. If you're using VMware vSphere, you can also use the AWS Connector for vCenter to export a VM from VMware and import it into Amazon EC2. For more information, see Migrating Your Virtual Machine to Amazon EC2 Using AWS Connector for vCenter in the *AWS Management Portal for vCenter User Guide*.

VM Import/Export can be used to migrate applications and workloads, copy your VM image catalog, or create a disaster recovery repository for VM images.

* **Migrate existing applications and workloads to Amazon EC2**—You can migrate your VM-based applications and workloads to Amazon EC2 and preserve their software and configuration settings. After you import your applications and workloads into an Amazon EC2 instance, you can create Amazon Machine Images (AMIs) and run multiple copies of the same image. You can also create snapshots and use them to back up your data. You can use AMI and snapshot copies to replicate your applications and workloads around the world. For more information about AMI copy, see Copying an AMI (p. 85).
* **Copy your VM image catalog to Amazon EC2**—You can copy your existing VM image catalog into Amazon EC2. If you maintain a catalog of approved VM images, you can copy your image catalog to Amazon EC2 and create Amazon EC2 instances from the imported VM images. Your existing software, including products that you have installed such as anti-virus software, intrusion detection systems, and so on, can be imported along with your VM images. You can use the resulting Amazon EC2 instances to create Amazon EC2 AMIs. You can use the AMIs as your image catalog within Amazon EC2.
* **Create a disaster recovery repository for VM images**—You can import your local VM images into Amazon EC2 for backup and disaster recovery purposes. You can store the imported images as Amazon Elastic Block Store (Amazon EBS)-backed AMIs so they're ready to launch in Amazon EC2 when you need them. If your local environment suffers an event, you can quickly launch your instances to preserve business continuity while simultaneously exporting them to rebuild your local infrastructure.

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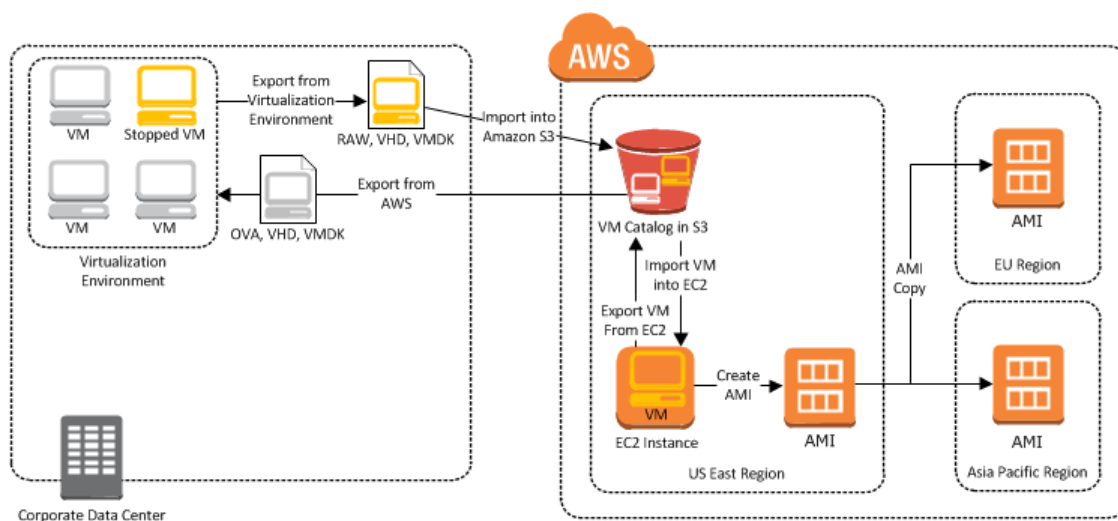
Contents

VM Import/Export Prerequisites

Before you begin the process of exporting an instance from your virtualization environment or importing and exporting a VM from Amazon EC2, you must be aware of the operating systems and image formats that AWS supports, and understand the limitations on exporting instances and volumes.

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

Contents

Operating Systems

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

Windows (32- and 64-bit)

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

Linux/Unix (64-bit)

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

    Note
    RHEL 6.0 is unsupported because it lacks the drivers required to run on Amazon EC2.
    VM Import supports license portability for RHEL instances. Your existing RHEL licenses are
    imported along with their associated RHEL instance. For more information about eligibility for
    Red Hat Cloud Access, see Eligibility at the Red Hat website.

- CentOS 5.1-5.10, 6.1-6.5

    Note
    CentOS 6.0 is unsupported because it lacks the drivers required to run on Amazon EC2.

- Ubuntu 12.04, 12.10, 13.04, 13.10
- Debian 6.0.0-6.0.8, 7.0.0-7.2.0

Image Formats

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

Importing Image Formats into Amazon EC2

AWS supports the following image formats for importing both volumes and instances into Amazon EC2:

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

> **Note**
> You can only import VMDK files into Amazon EC2 that were created through the OVF export process in VMware.

Exporting Image Formats from Amazon EC2

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

- Open Virtual Appliance (OVA) image format, which is compatible with VMware vSphere versions 4 and 5.
- Virtual Hard Disk (VHD) image format, which is compatible with Citrix Xen and Microsoft Hyper-V virtualization products.
- Stream-optimized ESX Virtual Machine Disk (VMDK) image format, which is compatible with VMware ESX and VMware vSphere versions 4 and 5 virtualization products.

Instance Types

AWS supports importing Windows instances into any instance type. Linux instances can be imported into the following instance types:

- General purpose: `t2.micro` | `t2.small` | `t2.medium` | `m3.medium` | `m3.large` | `m3.xlarge` | `m3.2xlarge`
- Compute optimized: `c3.large` | `c3.xlarge` | `c3.2xlarge` | `c3.4xlarge` | `cc2.8xlarge`
- Memory optimized: `cr1.8xlarge`
- Storage optimized: `hi1.4xlarge` | `hs1.8xlarge` | `i2.xlarge` | `i2.2xlarge` | `i2.4xlarge`
- GPU: `cg1.4xlarge`

Requirements and Limitations

Known Limitations for Importing a VM into Amazon EC2

Importing instances and volumes is subject to the following limitations:

- You can have up to five import tasks in progress at the same time per region.
- Imported instances create EC2 instances that use Hardware Virtual Machine (HVM) virtualization. Creating instances that use Paravirtual (PV) virtualization using VM Import is not supported. Linux PVHVM drivers are supported within imported instances.
- Imported Red Hat Enterprise Linux (RHEL) instances must use Cloud Access (BYOL) licenses.
- Imported Linux instances must use 64-bit images. Importing 32-bit Linux images is not supported.

- Imported Linux instances should use default kernels for best results. VMs that use custom Linux kernels might not import successfully.
- Typically, you import a compressed version of a disk image; the expanded image cannot exceed 1 TiB.
- Make sure your VM only uses a single disk. Importing a VM with more than one disk is not supported. For Linux VMs, /boot and / can be located in different partitions, but they need to be on the same disk.

  We suggest that you import the VM with only the boot volume, and import any additional disks using the ec2-import-volume command. After the `ImportInstance` task is complete, use the ec2-attach-volume command to associate the additional volumes with your instance.
- Make sure that you have at least 250 MB of available disk space for installing drivers and other software on any VM you want to import into an Amazon EC2 instance running Microsoft Windows.
- Imported instances automatically have access to the Amazon EC2 instance store, which is temporary disk storage from disks that are physically attached to the host computer. You cannot disable this during import. For more information about instance storage, see Amazon EC2 Instance Store (p. 581).
- Tasks must complete within 7 days of the start date.
- Multiple network interfaces are not currently supported. When converted and imported, your instance will have a single virtual NIC using DHCP for address assignment.
- Internet Protocol version 6 (IPv6) IP addresses are not supported.
- For vCenter 4.0 and vSphere 4.0 users, remove any attached CD-ROM images or ISOs from the virtual machine.
- Amazon VM Import does not install the single root I/O virtualization (SR-IOV) drivers on the c3 and i2 instance types, except for imports of Microsoft Windows Server 2012 R2 VMs. These drivers are not required unless you plan to use enhanced networking, which provides higher performance (packets per second), lower latency, and lower jitter. To enable enhanced networking on a c3 or i2 instance type after you import your VM, see Enabling Enhanced Networking on Linux Instances in a VPC (p. 507). For Microsoft Windows Server 2012 R2 VMs, SR-IOV driver are automatically installed as a part of the import process.

Known Limitations for Exporting a VM from Amazon EC2

Exporting instances and volumes is subject to the following limitations:

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

Importing a VM into Amazon EC2

There are two ways you can launch an instance in Amazon EC2. You can launch an instance from an Amazon Machine Image (AMI), or, you can launch an instance from a virtual machine (VM) that you imported from a virtualization environment such as Citrix Xen, Microsoft Hyper-V, or VMware vSphere. This section covers importing a VM and launching it as an Amazon EC2 instance. For more information about how to launch an Amazon EC2 instance from an AMI, see Launch Your Instance (p. 289).

To use your VM as an instance in Amazon EC2, you must first export it from the virtualization environment, and then import it to Amazon EC2 using the Amazon EC2 command line interface (CLI) or API tools. If you're importing a VM from VMware vCenter, you can also use the AWS Connector for vCenter to export a VM from VMware and import it into Amazon EC2. For more information, see Migrating Your Virtual Machine to Amazon EC2 Using AWS Connector for vCenter in the *AWS Management Portal for vCenter User Guide*.

Whether you use the CLI or the API, you will follow the same steps for importing VMs or volumes into Amazon EC2. This is the process for using the CLI.

To import a VM into Amazon EC2

1. Install the CLI. For more information, see Step 1: Install the Amazon EC2 CLI (p. 271).
2. Prepare the VM for import to Amazon EC2. For more information, see Step 2: Prepare Your VM (p. 272).
3. Export the VM from the virtualization environment. For more information, see Step 3: Export Your VM from Its Virtual Environment (p. 272).
4. Import the VM into Amazon EC2. For information, see Step 4: Importing Your VM into Amazon EC2 (p. 273).
5. Launch the instance in Amazon EC2. For more information, see Step 5: Launch the instance in Amazon EC2 (p. 279).

Step 1: Install the Amazon EC2 CLI

You need to install the Amazon EC2 CLI to import your Citrix, Microsoft Hyper-V, or VMware vSphere virtual machines into Amazon EC2 or to export them from Amazon EC2. If you haven't already installed the Amazon EC2 CLI, see Setting Up the Amazon EC2 Tools.

You'll use the following Amazon EC2 commands to import or export a VM.

Command	Description
ec2-import-instance	Creates a new import instance task using metadata from the specified disk image and imports the instance to Amazon EC2.
ec2-import-volume	Creates a new import volume task using metadata from the specified disk image and imports the volume to Amazon EC2.
ec2-resume-import	Resumes the upload of a disk image associated with an import instance or import volume task ID.
ec2-describe-conversion-tasks	Lists and describes your import tasks.
ec2-cancel-conversion-task	Cancels an active import task. The task can be the import of an instance or volume.
ec2-delete-disk-image	Deletes a partially or fully uploaded disk image for import from an Amazon S3 bucket.
ec2-create-image-export-task	Exports a running or stopped instance to an Amazon S3 bucket.
ec2-cancel-export-task	Cancels an active export task.
ec2-describe-export-tasks	Lists and describes your export tasks, including the most recent canceled and completed tasks.

For information about these commands and other Amazon EC2 commands, see the Amazon EC2 Command Line Reference.

Step 2: Prepare Your VM

Use the following guidelines to configure your VM before exporting it from the virtualization environment.

- Review the prerequisites. For more information, see VM Import/Export Prerequisites (p. 268).
- Disable any antivirus or intrusion detection software on your VM. These services can be re-enabled after the import process is complete.
- Uninstall the VMware Tools from your VMware VM.
- Disconnect any CD-ROM drives (virtual or physical).
- Set your network to DHCP instead of a static IP address. If you want to assign a static private IP address, be sure to use a non-reserved private IP address in your VPC subnet. Amazon Virtual Private Cloud (Amazon VPC) reserves the first four private IP addresses in a VPC subnet.
- Shut down your VM before exporting it.

Windows

- Enable Remote Desktop (RDP) for remote access.
- Make sure that your host firewall (Windows firewall or similar), if configured, allows access to RDP. Otherwise, you will not be able to access your instance after the import is complete.
- Make sure that the administrator account and all other user accounts use secure passwords. All accounts must have passwords or the importation might fail.
- Make sure that your Windows VM has .NET Framework 3.5 installed, as required by Amazon Windows EC2Config Service.
- Do not run System Preparation (Sysprep) on your Windows VM images. We recommend that you import the image and then use the Amazon EC2 Config service to run Sysprep.
- Disable Autologon on your Windows VM.
- Make sure that there are no pending Microsoft updates, and that the computer is not set to install software when it reboots.

Linux

- Enable Secure Shell (SSH) for remote access.
- Make sure that your host firewall (such as Linux iptables) allows access to SSH. Otherwise, you will not be able to access your instance after the import is complete.
- Make sure that you have configured a non-root user to use public key-based SSH to access your instance after it is imported. The use of password-based SSH and root login over SSH are both possible, but not recommended. The use of public keys and a non-root user is recommended because it is more secure. VM Import will not configure an *ec2-user* account as part of the import process.
- Make sure that your Linux VM uses GRUB (GRUB legacy) or GRUB 2 as its bootloader.
- Make sure that your Linux VM uses a root filesystem is one of the following: EXT2, EXT3, EXT4, Btrfs, JFS, or XFS.

Step 3: Export Your VM from Its Virtual Environment

After you have prepared your VM for export, you can export it from your virtualization environment. For information about how to export a VM from your virtualization environment, see the documentation for Citrix, Microsoft Hyper-V, or VMware vCenter virtualization environment.

Citrix: For more information, see Export VMs as OVF/OVA at the Citrix website.

Microsoft Hyper-V: For more information, see Hyper-V - Export & Import at the Microsoft website.

VMware: For more information, see Export an OVF Template at the VMware website.

Step 4: Importing Your VM into Amazon EC2

After exporting your VM from your virtualization environment, you can import it into Amazon EC2. The import process is the same regardless of the origin of the VM.

Here are some important things to know about your VM instance, as well as some security and storage recommendations:

- Amazon EC2 automatically assigns a DHCP IP address to your instance. The DNS name and IP address are available through the `ec2-describe-instances` command when the instance starts running.
- Your instance has only one Ethernet network interface.
- To specify a subnet to use when you create the import task, use the `--subnet` *subnet_id* option with the `ec2-import-instance` command; otherwise, your instance will use a public IP address. We recommend that you use a restrictive security group to control access to your instance.
- We recommend that your Windows instances contain strong passwords for all user accounts. We recommend that your Linux instances use public keys for SSH.
- For Windows instances, we recommend that you install the Amazon Windows EC2Config Service after you import your virtual machine into Amazon EC2.

To import a VM into Amazon EC2

Use ec2-import-instance to create a new import instance task.

The syntax of the command is as follows:

```
ec2-import-instance disk_image_filename -f file_format -t instance_type -a ar
chitecture -b s3_bucket_name -o owner -w secret_key -p platform_name
```

If the import of the VM is interrupted, you can use the `ec2-resume-import` command to resume the import from where it stopped. For more information, see Resuming an Upload (p. 277).

Example (Windows)

The following command creates an import instance task that imports a Windows Server 2008 SP2 (32-bit) VM.

```
C:\> ec2-import-instance ./WinSvr8-2-32-disk1.vmdk  -f VMDK -t m1.small -a i386
 -b myawsbucket -o AKIAIOSFODNN7EXAMPLE -w wJalrXUtnFEMI/K7MDENG/bPxRfi
CYEXAMPLEKEY -p Windows
```

This request uses the VMDK file, `WinSvr8-2-32-disk1.vmdk`, to create the import task. (Note that you can alternatively use VHD or RAW format.) If you do not specify a size for the requesting volume using the `-s` parameter, a volume size based on the disk image file is used. The output is similar to the following.

```
Requesting volume size: 25 GB
Disk image format: Stream-optimized VMDK
Converted volume size: 26843545600 bytes (25.00 GiB)
Requested EBS volume size: 26843545600 bytes (25.00 GiB)
TaskType        IMPORTINSTANCE  TaskId  import-i-fhbx6hua        ExpirationTime
  2011-09-09T15:03:38+00:00       Status  active  StatusMessage   Pending In
stanceID        i-6ced060c
```

```
DISKIMAGE        DiskImageFormat VMDK    DiskImageSize   5070303744
VolumeSize       25      AvailabilityZone        us-east-1c      Approximate
BytesConverted        0        Status   active StatusMessage    Pending
Creating new manifest at testImport/9cba4345-b73e-4469-8106-
2756a9f5a077/Win_2008_R1_EE_64.vmdkmanifest.xml
Uploading the manifest file
Uploading 5070303744 bytes across 484 parts
0% |-------------------------------------------------| 100%
    |=================================================|
Done
```

Example (Linux)

The following example creates an import instance task that imports a 64-bit Linux VM.

```
$ ec2-import-instance rhel6.4-64bit-disk.vhd -f vhd -t m3.xlarge -a x86_64 -b
myawsbucket -o AKIAIOSFODNN7EXAMPLE –w wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
 -p Linux
```

This request uses the VHD file, **rhel6.4-64bit-disk.vhd**, to create the import task. The output is similar to the following.

```
Requesting volume size: 8 GB
TaskType        IMPORTINSTANCE  TaskId  import-i-ffnzq636        ExpirationTime
  2013-12-12T22:55:18Z    Status        active StatusMessage   Pending InstanceID
      i-a56ab6dd
DISKIMAGE        DiskImageFormat VHD     DiskImageSize   861055488
VolumeSize       8       AvailabilityZone        us-east-1d      ApproximateBytesCon
verted        0        Status   active  StatusMessage   Pending
Creating new manifest at myawsbucket/b73bae14-7ec5-4122-8958-
4234028e1d9f/rhel6.4-64bit-disk.vhdmanifest.xml
Uploading the manifest file
Uploading 861055488 bytes across 83 parts
0% |-------------------------------------------------| 100%
    |=================================================|
Done

Average speed was 11.054 MBps

The disk image for import-i-ffnzq636 has been uploaded to Amazon S3 where it
is being converted into
an EC2 instance. You may monitor the progress of this task by running ec2-de
scribe-conversion-tasks.
When the task is completed, you may use ec2-delete-disk-image to remove the
image from S3.
```

Checking on the Status of Your Import Task

The ec2-describe-conversion-tasks command returns the status of an import task. Status values include the following:

- **active—**Your instance or volume is still importing.
- **cancelling—**Your instance or volume is still being canceled.
- **cancelled—**Your instance or volume is canceled.

- **completed—**Your instance or volume is ready to use.

  The imported instance is in the stopped state. You use `ec2-start-instance` to start it. For more information, see ec2-start-instances in the *Amazon EC2 Command Line Reference*.

To check the status of your import task

Use ec2-describe-conversion-tasks to return the status of the task. The syntax of the command is as follows:

```
ec2-describe-conversion-tasks task_id
```

Example

The following example enables you to see the status of your import instance task.

```
$ ec2-describe-conversion-tasks import-i-ffvko9js
```

Response 1

The following response shows that the `IMPORTINSTANCE` status is `active`, and 73747456 bytes out of 893968896 have been converted.

```
TaskType          IMPORTINSTANCE  TaskId  import-i-ffvko9js      ExpirationTime
   2011-06-07T13:30:50+00:00        Status   active  StatusMessage    Pending In
stanceID       i-17912579
DISKIMAGE        DiskImageFormat VMDK    DiskImageSize    893968896 VolumeSize
    12      AvailabilityZone       us-east-1      ApproximateBytesConverted
    73747456      Status   active StatusMessage    Pending
```

Response 2

The following response shows that the `IMPORTINSTANCE` status is `active`, at 7% progress, and the DISKIMAGE is completed.

```
TaskType          IMPORTINSTANCE  TaskId  import-i-ffvko9js      ExpirationTime
   2011-06-07T13:30:50+00:00        Status   active  StatusMessage    Progress: 7%
    InstanceID      i-17912579
DISKIMAGE        DiskImageFormat VMDK    DiskImageSize    893968896 VolumeId
    vol-9b59daf0    VolumeSize      12      AvailabilityZone        us-east-1
    ApproximateBytesConverted      893968896 Status   completed
```

Response 3

The following response shows that the `IMPORTINSTANCE` status is `completed`.

```
TaskType          IMPORTINSTANCE  TaskId  import-i-ffvko9js      ExpirationTime
   2011-06-07T13:30:50+00:00       Status   completed    InstanceID      i-17912579
DISKIMAGE        DiskImageFormat VMDK    DiskImageSize    893968896 VolumeId
    vol-9b59daf0    VolumeSize      12      AvailabilityZone        us-east-1
    ApproximateBytesConverted      893968896 Status   completed
```

Note
The `IMPORTINSTANCE` status is what you use to determine the final status. The `DISKIMAGE` status will be `completed` for a period of time before the `IMPORTINSTANCE` status is `completed`.

You can now use commands such as `ec2-stop-instance`, `ec2-start-instance`, `ec2-reboot-instance`, and `ec2-terminate-instance` to manage your instance. For more information, see the Amazon EC2 Command Line Reference

Importing Your Volumes into Amazon EBS

This section describes how to import your data storage into Amazon EBS, and then attach it to one of your existing EC2 instances. Amazon EC2 supports importing RAW, Virtual Hard Disk (VHD), and ESX Virtual Machine Disk (VMDK) disk formats.

Important
We recommend using Amazon EC2 security groups to limit network access to your imported instance. Configure a security group to allow only trusted EC2 instances and remote hosts to connect to RDP and other service ports. For more information about security groups, see Amazon EC2 Security Groups (p. 421).

After you have exported your virtual machine from the virtualization environment, importing the volume to Amazon EBS is a single-step process. You create an import task and upload the volume.

To import a volume into Amazon EBS

1. Use ec2-import-volume to create a task that allows you to upload your volume into Amazon EBS. The syntax of the command is as follows:

```
ec2-import-volume disk_image -f file_format -s volume_size -z availability_zone -b s3_bucket_name -o owner -w secret_key
```

The following example creates an import volume task for importing a volume to the us-east-1 region in the d availability zone.

```
$ ec2-import-volume Win_2008_R1_EE_64.vmdk -f vmdk -s 25 -z us-east-1d -b
myawsbucket -o AKIAIOSFODNN7EXAMPLE -w wJalrXUtnFEMI/K7MDENG/bPxRfi
CYEXAMPLEKEY --region us-east-1 -o AKIAI44QH8DHBEXAMPLE -w je7MtGbCl
wBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
```

The following is an example response.

```
Requesting volume size: 25 GB
Disk image format: Stream-optimized VMDK
Converted volume size: 26843545600 bytes (25.00 GiB)
Requested EBS volume size: 26843545600 bytes (25.00 GiB)
TaskType        IMPORTVOLUME    TaskId import-vol-ffut5xv4    ExpirationTime
   2011-09-09T15:22:30+00:00        Status    active  StatusMessage   Pending
DISKIMAGE       DiskImageFormat VMDK    DiskImageSize   5070303744
VolumeSize      25      AvailabilityZone        us-east-1d      Approximate
BytesConverted       0
Creating new manifest at myawsbucket/0fd8fcf5-04d8-44ae-981f-
3c9f56d04520/Win_2008_R1_EE_64.vmdkmanifest.xml
Uploading the manifest file
Uploading 5070303744 bytes across 484 parts
0% |--------------------------------------------| 100%
```

```
     |==================================================|
Done
```

Amazon EC2 returns a task ID that you use in the next step. In this example, the ID is `import-vol-ffut5xv4`.

2. Use ec2-describe-conversion-tasks to confirm that your volume imported successfully.

```
$ ec2-describe-conversion-tasks import-vol-ffut5xv4
TaskType          IMPORTVOLUME    TaskId  import-vol-ffut5xv4     ExpirationTime
   2011-09-09T15:22:30+00:00         Status  completed
DISKIMAGE          DiskImageFormat VMDK   DiskImageSize   5070303744
VolumeId          vol-365a385c    VolumeSize      25      AvailabilityZone
       us-east-1d      ApproximateBytesConverted       5070303744
```

The status in this example is `completed`, which means the import succeeded.

3. Use ec2-attach-volume to attach the Amazon EBS volume to one of your existing EC2 instances. The following example attaches the volume, `vol-2540994c`, to the `i-a149ec4a` instance on the device, `/dev/sde`.

```
$ 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 -t task_id -o owner -w secret_key
```

Example

The following example resumes an import instance task.

```
$ ec2-resume-import Win_2008_R1_EE_64.vmdk -t import-i-ffni8aei -o AKIAIOSFOD
NN7EXAMPLE -w wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

The following shows the output when the import instance task is complete:

```
Disk image size: 5070303744 bytes (4.72 GiB)
Disk image format: Stream-optimized VMDK
Converted volume size: 26843545600 bytes (25.00 GiB)
Requested EBS volume size: 26843545600 bytes (25.00 GiB)
Uploading 5070303744 bytes across 484 parts
0% |------------------------------------------------| 100%
```

```
   |==================================================|
Done
Average speed was 10.316 MBps
The disk image for import-i-ffni8aei has been uploaded to Amazon S3
where it is being converted into an EC2 instance.  You may monitor the
progress of this task by running ec2-describe-conversion-tasks.  When
the task is completed, you may use ec2-delete-disk-image to remove the
image from S3.
```

Canceling an Upload

Use ec2-cancel-conversion-task to cancel an active import task. The task can be the upload of an instance or a volume. The command removes all artifacts of the import, including uploaded volumes or instances.

If the import is complete or still transferring the final disk image, the command fails and returns an exception similar to the following:

```
Client.CancelConversionTask Error: Failed to cancel conversion task import-i-
fh95npoc
```

To cancel an upload task

Use the task ID of the upload you want to delete with ec2-cancel-conversion-task.

Example

The following example cancels the upload associated with the task ID `import-i-fh95npoc`.

```
$ ec2-cancel-conversion-task import-i-fh95npoc
```

The output for a successful cancellation is similar to the following:

```
CONVERSION-TASK import-i-fh95npoc
```

You can use the ec2-describe-conversion-tasks command to check the status of the cancellation as in the following example:

```
$ ec2-describe-conversion-tasks import-i-fh95npoc
TaskType        IMPORTINSTANCE  TaskId import-i-fh95npoc        ExpirationTime
  2010-12-20T18:36:39+00:00      Status  cancelled  InstanceID      i-825063ef
DISKIMAGE        DiskImageFormat VMDK    DiskImageSize   2671981568
VolumeSize      40       AvailabilityZone       us-east-1c ApproximateBytesCon
verted       0        Status  cancelled
```

In this example, the status is `cancelled`. If the upload were still in process, the status would be `cancelling`.

Cleaning Up After an Upload

You can use ec2-delete-disk-image to remove the image file after it is uploaded. If you do not delete it, you will be charged for its storage in Amazon S3.

To delete a disk image

Use the task ID of the disk image you want to delete with `ec2-delete-disk-image`.

Example

The following example deletes the disk image associated with the task ID, `import-i-fh95npoc`.

```
$ ec2-delete-disk-image -t import-i-fh95npoc
```

The output for a successful cancellation is similar to the following:

```
DELETE-TASK import-i-fh95npoc
```

Step 5: Launch the instance in Amazon EC2

After you upload the VM to Amazon S3, the VM Import process automatically converts it into an Amazon EC2 instance and launches it as a stopped instance in the Amazon EC2 console. Before you can begin using the instance, you must start it. For more information about working with an Amazon EC2 instance, see Instance Lifecycle (p. 286).

To start the instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see Regions and Endpoints.
3. In the navigation pane, click **Instances**.
4. In the content pane, right-click the instance, and then click **Start**.

Exporting Amazon EC2 Instances

If you have previously imported an instance into Amazon EC2, you can use the command line tools to export that instance to Citrix Xen, Microsoft Hyper-V, or VMware vSphere. Exporting an instance that you previously imported is useful when you want to deploy a copy of your EC2 instance in your on-site virtualization environment.

Contents

- Export an Instance (p. 279)
- Cancel or Stop the Export of an Instance (p. 280)

Export an Instance

You can use the Amazon EC2 CLI to export an instance. If you haven't installed the CLI already, see Setting Up the Amazon EC2 Tools.

The ec2-create-instance-export-task command gathers all of the information necessary (e.g., instance ID; name of the Amazon S3 bucket that will hold the exported image; name of the exported image; VMDK, OVA, or VHD format) to properly export the instance to the selected virtualization format. The exported file is saved in the Amazon S3 bucket that you designate.

> **Note**
> When you export an instance, you are charged the standard Amazon S3 rates for the bucket where the exported VM is stored. In addition, a small charge reflecting temporary use of an Amazon EBS snapshot might appear on your bill. For more information about Amazon S3 pricing, see Amazon Simple Storage Service (S3) Pricing.

To export an instance

1. Create an Amazon S3 bucket for storing the exported instances. The Amazon S3 bucket must grant **Upload/Delete** and **View Permissions** access to the **vm-import-export@amazon.com** account. For more information, see Creating a Bucket and Editing Bucket Permissions in the *Amazon Simple Storage Service Console User Guide*.

2. At a command prompt, type the following command:

```
ec2-create-instance-export-task instance_id -e target_environment -f
disk_image_format -c container_format -b s3_bucket
```

`instance_id`
    The ID of the instance you want to export.
`target_environment`
    VMware, Citrix, or Microsoft.
`disk_image_format`
    VMDK for VMware or VHD for Microsoft Hyper-V and Citrix Xen.
`container_format`
    Optionally set to OVA when exporting to VMware.
`s3_bucket`
    The name of the Amazon S3 bucket to which you want to export the instance.

3. To monitor the export of your instance, at the command prompt, type the following command, where `task_id` is the ID of the export task:

```
ec2-describe-export-tasks task_id
```

Cancel or Stop the Export of an Instance

You can use the Amazon EC2 CLI to cancel or stop the export of an instance up to the point of completion. The ec2-cancel-export-task command removes all artifacts of the export, including any partially created Amazon S3 objects. If the export task is complete or is in the process of transferring the final disk image, the command fails and returns an error.

To cancel or stop the export of an instance

At the command prompt, type the following command, where `task_id` is the ID of the export task:

```
ec2-cancel-export-task task_id
```

Troubleshooting VM Import/Export

When importing or exporting a VM, most errors occur when you attempt to do something that isn't sup-ported. To avoid these errors, read VM Import/Export Prerequisites (p. 268) before you begin an import or an export.

Errors

*   AWS Error Code: InvalidParameter, AWS Error Message: Parameter disk-image-size=0 has an invalid format. (p. 281)

AWS Error Code: InvalidParameter, AWS Error Message: Parameter disk-image-size=0 has an invalid format.

The image format you used is not supported.

Resolution

Retry using one of the supported image formats: RAW, VHD, or VMDK.

Client.UnsupportedOperation: This instance has multiple volumes attached. Please remove additional volumes.

The VM has multiple attached disks.

Resolution

Detach the extra drives and try again. If you need the data on the other volumes, copy the data to the root volume and try to export the VM again.

ClientError: Footers not identical

You attempted to import a fixed or differencing VHD, or there was an error in creating the VHD.

Resolution

Export your VM again and retry importing it into Amazon EC2.

ClientError: Uncompressed data has invalid length.

The VMDK file is corrupted.

Resolution

You can try repairing or recreating the VMDK file, or use another one for your import.

ERROR: Bucket `<MyBucketName>` is not in the `<RegionName>` region, it's in `<RegionName>`.

The Amazon S3 bucket is not in the same region as the instance you want to import.

Resolution

Try adding the `--ignore-region-affinity` option, which ignores whether the bucket's region matches the region where the import task is created. You can also create an Amazon S3 bucket using the Amazon Simple Storage Service console and set the region to the region where you want to import the VM. Run the command again and specify the new bucket you just created.

ERROR: File uses unsupported compression algorithm 0.

The VMDK was created using OVA format instead of OVF format.

Resolution

Create the VMDK in OVF format.

Error starting instances: Invalid value `<instance ID>` for instanceId. Instance does not have a volume attached at root (/dev/sda1).

You attempted to start the instance before the VM import process and all conversion tasks were complete.

Resolution

Wait for the VM import process and all conversion tasks to completely finish, and then start the instance.

java.lang.OutOfMemoryError: Java heap space

There is not enough virtual memory available to launch Java, or the image you are trying to import is too large.

Resolution

If you allocate extra memory to Java, the extra memory will only apply to JVM, but if that setting is specified (explicitly for the EC2 command line tools) it will override the global settings. For example, you can use the following command to allocate 512 MB of extra memory to Java 'set EC2_JVM_ARGS=-Xmx512m'.

Service.InternalError: An internal error has occurred. Status Code: 500, AWS Service: AmazonEC2

You tried to import an instance that does not have a default VPC without specifying the subnet and Availability Zone.

Resolution

If you're importing an instance without a default VPC, be sure to specify the subnet and Availability Zone.

FirstBootFailure: This import request failed because the Windows instance failed to boot and establish network connectivity.

When you import a VM using the `ec2-import-instance` command, the import task might stop before its completed, and then fail. To investigate what went wrong, you can use the ec2-describe-conversion-tasks command to describe the instance.

When you receive the FirstBootFailure error message, it means that your virtual disk image was unable to perform one of the following steps:

- Boot up and start Windows.
- Install Amazon EC2 networking and disk drivers.
- Use a DHCP-configured network interface to retrieve an IP address.
- Activate Windows using the Amazon EC2 Windows volume license.

The following best practices can help you to avoid Windows first boot failures:

- **Disable anti-virus and anti-spyware software and firewalls**. These types of software can prevent installing new Windows services or drivers or prevent unknown binaries from running. Software and firewalls can be re-enabled after importing.
- **Do not harden your operating system**. Security configurations, sometimes called hardening, can prevent unattended installation of Amazon EC2 drivers. There are numerous Windows configuration settings that can prevent import. These settings can be reapplied once imported.
- **Disable or delete multiple bootable partitions**. If your virtual machine boots and requires you to choose which boot partition to use, the import may fail.

This inability of the virtual disk image to boot up and establish network connectivity could be due to any of the following causes.

Causes

The installation of Windows is not valid on the virtual machine

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

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

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

TCP/IP networking and DHCP are not enabled

Cause: For any Amazon EC2 instance, including those in Amazon VPC, TCP/IP networking and DHCP must be enabled. Within a VPC, you can define an IP address for the instance either before or after importing the instance. Do not set a static IP address before exporting the instance.

Resolution: Ensure that TCP/IP networking is enabled. For more information, see Setting up TCP/IP (Windows Server 2003) or Configuring TCP/IP (Windows Server 2008) at the Microsoft TechNet website.

Ensure that DHCP is enabled. For more information, see What is DHCP at the Microsoft TechNet web site.

A volume that Windows requires is missing from the virtual machine

Cause: Importing a VM into Amazon EC2 only imports the boot disk, all other disks must be detached and Windows must able to boot before importing the virtual machine. For example, Active Directory often stores the Active Directory database on the D:\ drive. A domain controller cannot boot if the Active Directory database is missing or inaccessible.

Resolution: Detach any secondary and network disks attached to the Windows VM before exporting.

Move any Active Directory databases from secondary drives or partitions onto the primary Windows partition. For more information, see "Directory Services cannot start" error message when you start your Windows-based or SBS-based domain controller at the Microsoft Support website.

Windows always boots into System Recovery Options

Cause: Windows can boot into System Recovery Options for a variety of reasons, including when Windows is pulled into a virtualized environment from a physical machine, also known as P2V.

Resolution: Ensure that Windows boots to a login prompt before exporting and preparing for import.

Do not import virtualized Windows instances that have come from a physical machine.

The virtual machine was created using a physical-to-virtual (P2V) conversion process

Cause: A P2V conversion occurs when a disk image is created by performing the Windows installation process on a physical machine and then importing a copy of that Windows installation into a VM. VMs that are created as the result of a P2V conversion are not supported by Amazon EC2 VM import. Amazon EC2 VM import only supports Windows images that were natively installed inside the source VM.

Resolution: Install Windows in a virtualized environment and migrate your installed software to that new VM.

Windows activation fails

Cause: During boot, Windows will detect a change of hardware and attempt activation. During the import process we attempt to switch the licensing mechanism in Windows to a volume license provided by Amazon Web Services. However, if the Windows activation process does not succeed, then the import will not succeed.

Resolution: Ensure that the version of Windows you are importing supports volume licensing. Beta or preview versions of Windows might not.

No bootable partition found

Cause: During the import process of a virtual machine, we could not find the boot partition.

Resolution: Ensure that the disk you are importing has the boot partition. We do not support multi-disk import.

Linux is not supported on the requested instance

Cause: Linux import is only supported on specific instance types. You attempted to import an unsupported instance type.

Resolution: Retry using one of the supported instance types.

- General purpose: `t2.micro` | `t2.small` | `t2.medium` | `m3.medium` | `m3.large` | `m3.xlarge` | `m3.2xlarge`
- Compute optimized: `c3.large` | `c3.xlarge` | `c3.2xlarge` | `c3.4xlarge` | `cc2.8xlarge`
- Memory optimized: `cr1.8xlarge`
- Storage optimized: `hi1.4xlarge` | `hs1.8xlarge` | `i2.xlarge` | `i2.2xlarge` | `i2.4xlarge`
- GPU: `cg1.4xlarge`

Instance Lifecycle

By working with Amazon EC2 to manage your instances from the moment you launch them through their termination, you ensure that your customers have the best possible experience with the applications or sites that you host on your instances.

The following illustration represents the transitions between instance states. Notice that you can't stop and start an instance store-backed instance. For more information about instance store-backed instances, see Storage for the Root Device (p. 55).
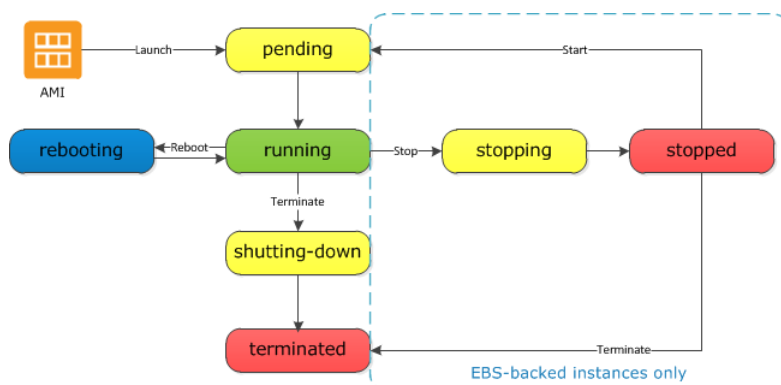
Instance Launch

When you launch an instance, it enters the `pending` state. The instance type that you specified at launch determines the hardware of the host computer for your instance. We use the Amazon Machine Image (AMI) you specified at launch to boot the instance. After the instance is ready for you, it enters the `running` state. You can connect to your running instance and use it the way that you'd use a computer sitting in front of you.

As soon as your instance starts to boot, you're billed for each hour or partial hour that you keep the instance running (even if the instance remains idle and you don't connect to it).

For more information, see Launch Your Instance (p. 289) and Connect to Your Linux Instance (p. 299).

Instance Stop and Start (Amazon EBS-backed instances only)

If your instance fails a status check or is not running your applications as expected, and if the root volume of your instance is an Amazon EBS volume, you can stop and start your instance to try to fix the problem.

When you stop your instance, it enters the `stopping` state, and then the `stopped` state. We don't charge hourly usage or data transfer fees for your instance after you stop it, but we do charge for the storage for any Amazon EBS volumes. While your instance is in the `stopped` state, you can modify certain attributes of the instance, including the instance type.

When you start your instance, it enters the `pending` state, and we move the instance to a new host computer. Therefore, when you stop and start your instance, you'll lose any data on the instance store volumes on the previous host computer.

If your instance is running in EC2-Classic, it receives a new private IP address, which means that an Elastic IP address (EIP) associated with the private IP address is no longer associated with your instance. If your instance is running in EC2-VPC, it retains its private IP address, which means that an EIP associated with the private IP address or network interface is still associated with your instance.

Each time you transition an instance from `stopped` to `running`, we charge a full instance hour, even if these transitions happen multiple times within a single hour.

For more information, see Stop and Start Your Instance (p. 309).

Instance Reboot

You can reboot your instance using the Amazon EC2 console, the Amazon EC2 CLI, and the Amazon EC2 API. We recommend that you use Amazon EC2 to reboot your instance instead of running the oper-ating system reboot command from your instance.

Rebooting an instance is equivalent to rebooting an operating system; the instance remains on the same host computer and maintains its public DNS name, private IP address, and any data on its instance store volumes. It typically takes a few minutes for the reboot to complete, but the time it takes to reboot depends on the instance configuration.

Rebooting an instance doesn't start a new instance billing hour.

For more information, see Reboot Your Instance (p. 312).

Instance Retirement

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

For more information, see Instance Retirement (p. 312).

Instance Termination

When you've decided that you no longer need an instance, you can terminate it. As soon as the status of an instance changes to `shutting-down` or `terminated`, you stop incurring charges for that instance.

Note that if you enable termination protection, you can't terminate the instance using the console, CLI, or API.

After you terminate an instance, it remains visible in the console for a short while, and then the entry is deleted. You can also describe a terminated instance using the CLI and API. You can't connect to or recover a terminated instance.

Each Amazon EBS-backed instance supports the `InstanceInitiatedShutdownBehavior` attribute, which controls whether the instance stops or terminates when you initiate a shutdown from within the instance itself (for example, by using the **shutdown** command on Linux). The default behavior is to stop the instance. You can modify the setting of this attribute while the instance is running or stopped.

Each Amazon EBS volume supports the `DeleteOnTermination` attribute, which controls whether the volume is deleted or preserved when you terminate the instance it is attached to. The default is to preserve volumes that you attach to a running instance and delete volumes that you attach at launch, such as the root volume.

For more information, see .

Differences Between Reboot, Stop, and Terminate

The following table summarizes the key differences between rebooting, stopping, and terminating your instance.

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Terminate
Host computer	The instance stays on the same host computer	The instance runs on a new host computer	None
Private and public IP addresses	These addresses stay the same	EC2-Classic: The instance gets new private and public IP addresses EC2-VPC: The instance keeps its private IP address. The instance gets a new public IP address, unless it has an Elastic IP address (EIP), which doesn't change during a stop/start.	None

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Terminate
Elastic IP addresses (EIP)	The EIP remains associated with the instance	EC2-Classic: The EIP is disassociated from the instance EC2-VPC: The EIP remains associated with the instance	The EIP is disassociated from the instance
Instance store volumes	The data is preserved	The data is erased	The data is erased
Root device volume	The volume is preserved	The volume is preserved	The volume is deleted by default
Billing	The instance billing hour doesn't change.	You stop incurring charges for an instance as soon as its state changes to `stopping`. Each time an instance transitions from `stopped` to `pending`, we start a new instance billing hour.	You stop incurring charges for an instance as soon as its state changes to `shutting-down`.

Note that operating system shutdown commands always terminate an instance store-backed instance. You can control whether operating system shutdown commands stop or terminate an Amazon EBS-backed instance. For more information, see Changing the Instance Initiated Shutdown Behavior (p. 317).

Launch Your Instance

An instance is a virtual server in the AWS cloud. You launch an instance from an Amazon Machine Image (AMI). The AMI provides the operating system, application server, and applications for your instance.

When you sign up for AWS, you can get started with Amazon EC2 for free using the AWS Free Usage Tier. You can either leverage the Free Usage Tier to launch and use a micro instance for free for 12 months. If you launch an instance that is not within the Free Usage Tier, you incur the standard Amazon EC2 usage fees for the instance. For more information, see the Amazon EC2 Pricing.

You can launch an instance using the following methods.

Method	Documentation
Use the Amazon EC2 console with an AMI that you select	Launching an Instance (p. 290)
Use the Amazon EC2 console to launch an instance using an existing instance as a template	Launching an Instance Using an Existing Instance as a Template (p. 295)
Use the Amazon EC2 console with an Amazon EBS snapshot that you created	Launching a Linux Instance from a Backup (p. 296)

Method	Documentation
Use the Amazon EC2 console with an AMI that you purchased from the AWS Marketplace	Launching an AWS Marketplace Instance (p. 297)
Use the AWS CLI with an AMI that you select	Using Amazon EC2 through the AWS CLI
Use the Amazon EC2 CLI with an AMI that you select	Launching an Instance Using the Amazon EC2 CLI
Use the AWS Tools for Windows PowerShell with an AMI that you select	Amazon EC2 from the AWS Tools for Windows PowerShell

After you launch your instance, you can connect to it and use it. To begin, the instance state is `pending`. When the instance state is `running`, the instance has started booting. There might be a short time before you can connect to the instance. The instance receives a public DNS name that you can use to contact the instance from the Internet. The instance also receives a private DNS name that other instances within the same Amazon EC2 network (EC2-Classic or EC2-VPC) can use to contact the instance. For more information about connecting to your instance, see Connect to Your Linux Instance (p. 299).

When you are finished with an instance, be sure to terminate it. For more information, see Terminate Your Instance (p. 315).

Launching an Instance

Before you launch your instance, be sure that you are set up. For more information, see Setting Up with Amazon EC2 (p. 20).

Your AWS account might support both the EC2-Classic and EC2-VPC platforms, depending on when you created your account and which regions you've used. To find out which platform your account supports, see Supported Platforms (p. 471). If your account supports EC2-Classic, you can launch an instance into either platform. If your account supports EC2-VPC only, you can launch an instance into a VPC only.

> **Important**
> When you launch an instance that's not within the AWS Free Usage Tier, you are charged for the time that the instance is running, even if it remains idle.

Launching Your Instance from an AMI

When you launch an instance, you must select a configuration, known as an Amazon Machine Image (AMI). An AMI contains the information required to create a new instance. For example, an AMI might contain the software required to act as a web server: for example, Linux, Apache, and your web site.

To launch an instance

1. Open the Amazon EC2 console.
2. In the navigation bar at the top of the screen, the current region is displayed. Select the region for the instance. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. Select the region that meets your needs. For more information, see Resource Locations (p. 605).
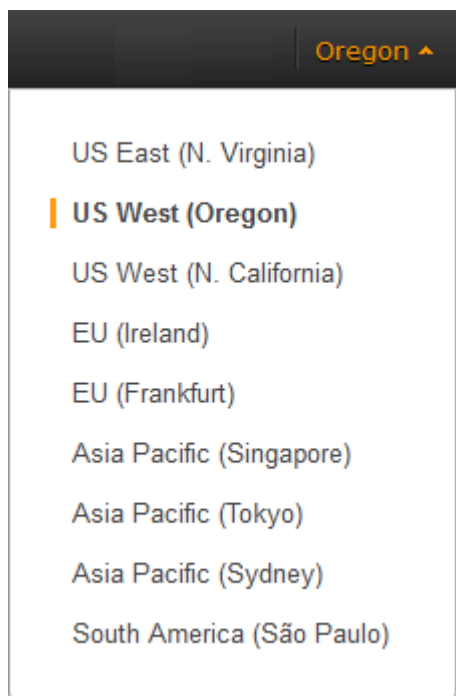
3. From the Amazon EC2 console dashboard, click **Launch Instance**.

4. On the **Choose an Amazon Machine Image (AMI)** page, choose an AMI as follows:

    a. Select the type of AMI to use in the left pane:

        Quick Start
          A selection of popular AMIs to help you get started quickly. To ensure that you select an AMI that is eligible for the free tier, click **Free tier only** in the left pane. (Notice that these AMIs are marked **Free tier eligible**.)

        My AMIs
          The private AMIs that you own, or private AMIs that have been shared with you.

        AWS Marketplace
          An online store where you can buy software that runs on AWS, including AMIs. For more information about launching an instance from the AWS Marketplace, see Launching an AWS Marketplace Instance (p. 297).

        Community AMIs
          The AMIs that AWS community members have made available for others to use. To filter the list of AMIs by operating system, select the **Amazon Linux** check box under **Operating system**. You can also filter by architecture and root device type.

    b. Check the **Root device type** listed for each AMI. Notice which AMIs are the type that you need, either `ebs` (backed by Amazon EBS) or `instance-store` (backed by instance store). For more information, see Storage for the Root Device (p. 55).

    c. Check the **Virtualization type** listed for each AMI. Notice which AMIs are the type that you need, either `hvm` or `paravirtual`. For example, some instance types require HVM. For more information, see Virtualization Types (p. 58).

    d. Choose an AMI that meets your needs, and then click **Select**.

5. On the **Choose an Instance Type** page, select the hardware configuration and size of the instance to launch. Larger instance types have more CPU and memory. For more information, see Instance Types (p. 103).

   To remain eligible for the free tier, select the **t2.micro** instance type. For more information, see T2 Instances (p. 105).

   By default, the wizard displays current generation instance types, and selects the first available instance type based on the AMI that you selected. To view previous generation instance types, select **All generations** from the filter list.

   > **Tip**
   > If you are new to AWS and would like to set up an instance quickly for testing purposes, you can click **Review and Launch** at this point to accept default configuration settings, and launch your instance. Otherwise, to configure your instance further, click **Next: Configure Instance Details**.

6. On the **Configure Instance Details** page, change the following settings as necessary (expand **Advanced Details** to see all the settings), and then click **Next: Add Storage**:

   - **Number of instances**: Enter the number of instances to launch.
   - **Purchasing option**: Select **Request Spot Instances** to launch a Spot Instance. For more information, see Spot Instances (p. 132).
   - Your account may support the EC2-Classic and EC2-VPC platforms, or EC2-VPC only. To find out which platform your account supports, see Supported Platforms (p. 471). If your account supports EC2-VPC only, you can launch your instance into your default VPC or a nondefault VPC. Otherwise, you can launch your instance into EC2-Classic or a nondefault VPC.

     > **Note**
     > You must launch a T2 instance into a VPC. If you don't have a VPC, you can let the wizard create one for you.

   To launch into EC2-Classic:
   - **Network**: Select **Launch into EC2-Classic**.
   - **Availability Zone**: Select the Availability Zone to use. To let AWS choose an Availability Zone for you, select **No preference**.

   To launch into a VPC:
   - **Network**: Select the VPC, or to create a new VPC, click **Create new VPC** to go the Amazon VPC console. When you have finished, return to the wizard and click **Refresh** to load your VPC in the list.
   - **Subnet**: Select the subnet into which to launch your instance. If your account is EC2-VPC only, select **No preference** to let AWS choose a default subnet in any Availability Zone. To create a new subnet, click **Create new subnet** to go to the Amazon VPC console. When you are done, return to the wizard and click **Refresh** to load your subnet in the list.
   - **Auto-assign Public IP**: Specify whether your instance receives a public IP address. By default, instances in a default subnet receive a public IP address and instances in a nondefault subnet do not. You can select **Enable** or **Disable** to override the subnet's default setting. For more information, see Public IP Addresses and External DNS Hostnames (p. 481).
   - **IAM role**: If applicable, select an AWS Identity and Access Management (IAM) role to associate with the instance. For more information, see IAM Roles for Amazon EC2 (p. 461).
   - **Shutdown behavior**: Select whether the instance should stop or terminate when shut down. For more information, see Changing the Instance Initiated Shutdown Behavior (p. 317).
   - **Enable termination protection**: Select this check box to prevent accidental termination. For more information, see Enabling Termination Protection for an Instance (p. 316).

- **Monitoring**: Select this check box to enable detailed monitoring of your instance using Amazon CloudWatch. Additional charges apply. For more information, see Monitoring Your Instances with CloudWatch (p. 354).

- **EBS-Optimized instance**: An Amazon EBS-optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. If the instance type supports this feature, select this check box to enable it. Additional charges apply. For more information, see Amazon EBS–Optimized Instances (p. 125).

- **Tenancy**: If you are launching your instance into a VPC, you can select **Dedicated tenancy** to run your instance on isolated, dedicated hardware. Additional charges apply. For more information, see Dedicated Instances in the *Amazon VPC User Guide*.

- **Network interfaces**: If you are launching an instance into a VPC and you did not select **No Preference** for your subnet, you can specify up to two network interfaces in the wizard. Click **Add IP** to assign more than one IP address to the selected interface. For more information about network interfaces, see Elastic Network Interfaces (ENI) (p. 494). If you selected the **Public IP** check box above, you can only assign a public IP address to a single, new network interface with the device index of eth0. For more information, see Assigning a Public IP Address (p. 484).

- **Kernel ID**: (Only valid for paravirtual (PV) AMIs) Select **Use default** unless you want to use a specific kernel.

- **RAM disk ID**: (Only valid for paravirtual (PV) AMIs) Select **Use default** unless you want to use a specific RAM disk. If you have selected a kernel, you may need to select a specific RAM disk with the drivers to support it.

- **Placement group**: A placement group is a logical grouping for your cluster instances. Select an existing placement group, or create a new one. This option is only available if you've selected an instance type that supports placement groups. For more information, see Placement Groups (p. 126).

- **User data**: You can specify user data to configure an instance during launch, or to run a configuration script. To attach a file, select the **As file** option and browse for the file to attach.

7. On the **Add Storage** page, you can specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume). You can change the following options, then click **Next: Tag Instance** when you have finished:

   - **Type**: Select instance store or Amazon EBS volumes to associate with your instance. The type of volume available in the list depends on the instance type you've chosen. For more information, see Amazon EC2 Instance Store (p. 581) and Amazon EBS Volumes (p. 521).

   - **Device**: Select from the list of available device names for the volume.

   - **Snapshot**: Enter the name or ID of the snapshot from which to restore a volume. You can also search for public snapshots by typing text into the **Snapshot** field. Snapshot descriptions are case-sensitive.

   - **Size**: For Amazon EBS-backed volumes, you can specify a storage size. Note that even if you have selected an AMI and instance that are eligible for the free usage tier, you need to keep under 30 GiB of total storage to stay within the free usage tier.

     Note
     If you increase the size of your root volume at this point (or any other volume created from a snapshot), you need to extend the file system on that volume in order to use the extra space. For more information about extending your file system after your instance has launched, see Expanding the Storage Space of a Volume (p. 545).

   - **Volume Type**: For Amazon EBS volumes, select either a General Purpose (SSD), Provisioned IOPS (SSD), or Magnetic volume. For more information, see Amazon EBS Volume Types (p. 523).

     Note
     If you select a Magnetic boot volume, you'll be prompted when you complete the wizard to make General Purpose (SSD) volumes the default boot volume for this instance and future console launches. (This preference persists in the browser session, and does not affect AMIs with Provisioned IOPS (SSD) boot volumes.) We recommended that you

make General Purpose (SSD) volumes the default because they provide a much faster boot experience and they are the optimal volume type for most workloads. For more information, see Amazon EBS Volume Types (p. 523).

Note
Some AWS accounts created before 2012 might have access to Availability Zones in us-east-1, us-west-1, or ap-northeast-1 that do not support SSD volumes such as Provisioned IOPS (SSD) and General Purpose (SSD). If you are unable to create an SSD volume (or launch an instance with an SSD volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports General Purpose (SSD) and Provisioned IOPS (SSD) volumes by creating a 1 GiB General Purpose (SSD) volume in that zone.

- **IOPS**: If you have selected a Provisioned IOPS (SSD) volume type, then you can enter the number of I/O operations per second (IOPS) that the volume can support.
- **Delete on Termination**: For Amazon EBS volumes, select this check box to delete the volume when the instance is terminated. For more information, see Preserving Amazon EBS Volumes on Instance Termination (p. 318).
- **Encrypted**: Select this check box to encrypt new Amazon EBS volumes. Amazon EBS volumes that are restored from encrypted snapshots are automatically encrypted. Encrypted volumes may only be attached to supported instance types (p. 565).

  Note
  Encrypted boot volumes are not supported at this time.

8. On the **Tag Instance** page, specify tags (p. 610) for the instance by providing key and value combinations. Click **Create Tag** to add more than one tag to your resource. Click **Next: Configure Security Group** when you are done.

9. On the **Configure Security Group** page, use a security group to define firewall rules for your instance. These rules specify which incoming network traffic is delivered to your instance. All other traffic is ignored. (For more information about security groups, see Amazon EC2 Security Groups (p. 421).) Select or create a security group as follows, and then click **Review and Launch**.

   To select an existing security group:

   1. Click **Select an existing security group**. Your security groups are displayed. (If you are launching into EC2-Classic, these are security groups for EC2-Classic. If you are launching into a VPC, these are security group for that VPC.)
   2. Select a security group from the list.
   3. (Optional) You can't edit the rules of an existing security group, but you can copy them to a new group by clicking **Copy to new**. Then you can add rules as described in the next procedure.

   To create a new security group:

   1. Click **Create a new security group**. The wizard automatically defines the launch-wizard-*x* security group.
   2. (Optional) You can edit the name and description of the security group.
   3. The wizard automatically defines an inbound rule to allow to you connect to your instance over SSH (port 22).

      Caution
      This rule enables all IP addresses (`0.0.0.0/0`) to access your instance over the specified port. This is acceptable for this short exercise, but it's unsafe for production environments. You should authorize only a specific IP address or range of addresses to access your instance.

4. You can add rules to suit your needs. For example, if your instance is a web server, open ports 80 (HTTP) and 443 (HTTPS) to allow Internet traffic.

    To add a rule, click **Add Rule**, select the protocol to open to network traffic, and then specify the source. Select **My IP** from the **Source** list to let the wizard add your computer's public IP address. However, if you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

10. On the **Review Instance Launch** page, check the details of your instance, and make any necessary changes by clicking the appropriate **Edit** link.

    When you are ready, click **Launch**.

11. In the **Select an existing key pair or create a new key pair** dialog box, you can choose an existing key pair, or create a new one. For example, select **Choose an existing key pair**, then select the key pair you created when getting set up.

    To launch your instance, select the acknowledgment check box, then click **Launch Instances**.

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

12. If the instance state immediately goes to `terminated` instead of `running`, you can get information about why the instance didn't launch. For more information, see .

Launching an Instance Using an Existing Instance as a Template

The Amazon EC2 console provides a **Launch More Like This** wizard option that enables you to use a current instance as a template for launching other instances. This option automatically populates the Amazon EC2 launch wizard with certain configuration details from the selected instance.

> **Note**
> The **Launch More Like This** wizard option does not clone your selected instance; it only replicates some configuration details. To create a copy of your instance, first create an AMI from it, then launch more instances from the AMI.

The following configuration details are copied from the selected instance into the launch wizard:

- AMI ID
- Instance type
- Availability Zone, or the VPC and subnet in which the selected instance is located
- Tags associated with the instance, if applicable
- Kernel ID and RAM disk ID, if applicable
- IAM role associated with the instance, if applicable
- Security group associated with the instance
- Tenancy setting, if launching into a VPC (shared or dedicated)
- Amazon EBS-optimization setting (true or false)
- Public IP address. If the selected instance currently has a public IP address, the new instance receives a public IP address - regardless of the selected instance's default public IP address setting. For more information about public IP addresses, see .

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

- Storage: The default storage configuration is determined by the AMI and the instance type.
- Termination protection: Disabled by default.
- Shutdown behavior: Set to 'stop' by default.
- User data: None by default.

To use your current instance as a template

1. On the Instances page, select the instance you want to use.
2. Click **Actions**, and select **Launch More Like This**.
3. The launch wizard opens on the **Review Instance Launch** page. You can check the details of your instance, and make any necessary changes by clicking the appropriate **Edit** link.

   When you are ready, click **Launch** to select a key pair and launch your instance.

Launching a Linux Instance from a Backup

With an Amazon EBS-backed Linux instance, you can back up the root device volume of the instance by creating a snapshot. When 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. This can be useful if you don't have the original AMI that you launched an instance from, but you need to be able to launch an instance using the same image.

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

Use the following procedure to create an AMI from the root volume of your instance using the console. If you prefer, you can use the register-image (AWS CLI) or ec2-register (Amazon EC2 CLI) command instead.

To create an AMI from your root volume using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, under **Elastic Block Store**, click **Snapshots**.
3. Click **Create Snapshot**.
4. In the **Volumes** field, start typing the name or ID of the root volume, and then select it from the list of options.
5. Select the snapshot that you just created, and then select **Create Image** from the **Actions** list.
6. In the **Create Image from EBS Snapshot** dialog box, complete the fields to create your AMI, then click **Create**. Be sure to do the following:

   - Select the architecture from the **Architecture** list (**i386** for 32-bit or **x86_64** for 64-bit).
   - Select the AKI from the **Kernel ID** list. If you select the default AKI or don't select an AKI, you'll be required to specify an AKI every time you launch an instance. In addition, your instance may fail the health check because the default AKI is incompatible with the instance.

7. In the navigation pane, select **AMIs**.
8. Select the AMI that you just created, and then click **Launch**. Follow the wizard to launch your instance. For more information about how to configure each step in the wizard, see Launching an Instance (p. 290).

Launching an AWS Marketplace Instance

You can subscribe to an AWS Marketplace product and launch an instance from the product's AMI using the Amazon EC2 launch wizard. For more information about paid AMIs, see Paid AMIs (p. 70). To cancel your subscription after launch, you first have to terminate all instances running from it. For more information, see Managing Your AWS Marketplace Subscriptions (p. 73).

To launch an instance from the AWS Marketplace using the launch wizard

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the Amazon EC2 dashboard, click **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, select the **AWS Marketplace** category on the left. Find a suitable AMI by browsing the categories, or using the search functionality. Click **Select** to choose your product.
4. A dialog displays an overview of the product you've selected. You can view the pricing information, as well as any other information that the vendor has provided. When you're ready, click **Continue**.

    > **Note**
    > You are not charged for using the product until you have launched an instance with the AMI. Take note of the pricing for each supported instance type, as you will be prompted to select an instance type on the next page of the wizard.

5. On the **Choose an Instance Type** page, select the hardware configuration and size of the instance to launch. When you're done, click **Next: Configure Instance Details**.
6. On the next pages of the wizard, you can configure your instance, add storage, and add tags. For more information about the different options you can configure, see Launching an Instance (p. 290). Click **Next** until you reach the **Configure Security Group** page.

    The wizard creates a new security group according to the vendor's specifications for the product. The security group may include rules that allow all IP addresses (`0.0.0.0/0`) access on SSH (port 22). We recommend that you adjust these rules to allow only a specific IP address or range of addresses to access your instance over those specific ports.

    When you are ready, click **Review and Launch**.
7. On the **Review Instance Launch** page, check the details of the AMI from which you're about to launch the instance, as well as the other configuration details you set up in the wizard. When you're ready, click **Launch** to choose or create a key pair, and launch your instance.
8. Depending on the product you've subscribed to, the instance may take a few minutes or more to launch. You are first subscribed to the product before your instance can launch. If there are any problems with your credit card details, you will be asked to update your account details. When the launch confirmation page displays, click **View Instances** to go to the Instances page.

    > **Note**
    > You are charged the subscription price as long as your instance is running, even if it is idle. If your instance is stopped, you may still be charged for storage.

9. When your instance is in the **running** state, you can connect to it. To do this, select your instance in the list and click **Connect**. Follow the instructions in the dialog. For more information about connecting to your instance, see Connect to Your Linux Instance (p. 299).

    > **Important**
    > Check the vendor's usage instructions carefully, as you may need to use a specific user name to log in to the instance. For more information about accessing your subscription details, see Managing Your AWS Marketplace Subscriptions (p. 73).

Launching an AWS Marketplace AMI Instance Using the API and CLI

To launch instances from AWS Marketplace products using the API or command line tools, first ensure that you are subscribed to the product. You can then launch an instance with the product's AMI ID using the following methods:

Method	Documentation
AWS CLI	Use the run-instances command, or see the following topic for more information: Launching an Instance.
Amazon EC2 CLI	Use the ec2-run-instances command, or see the following topic for more information: Launching an Instance Using the Amazon EC2 CLI.
AWS Tools for Windows PowerShell	Use the New-EC2Instance command, or see the following topic for more information: Launch an Amazon EC2 Instance Using Windows PowerShell
Query API	Use the RunInstances request.

Connect to Your Linux Instance

Learn how to connect to the instances that you launched and transfer files between your local computer and your instance.

Your Computer	Topic
Linux	Connecting to Your Linux Instance Using SSH (p. 299)
Windows	Connecting to Your Linux Instance from Windows Using PuTTY (p. 303)
All	Connecting to Your Linux Instance Using MindTerm (p. 308)

After you connect to your instance, you can try one of our tutorials, such as Tutorial: Installing a LAMP Web Server (p. 38) or Tutorial: Hosting a WordPress Blog with Amazon EC2 (p. 45).

Connecting to Your Linux Instance Using SSH

After you launch your instance, you can connect to it and use it the way that you'd use a computer sitting in front of you.

> **Note**
> It may take a couple of minutes after launch for your running instance to finish provisioning so that you can connect to it. Check that your instance has passed its status checks - you can view this information in the **Status Checks** column on the **Instances** page. If you receive an error while attempting to connect to your instance, see Troubleshooting Connecting to Your Instance.

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

Prerequisites

- **Install an SSH client**
  Your Linux computer most likely includes 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.

- **Install the Amazon EC2 CLI Tools**
  (Optional) If you're using a public AMI from a third party, you can use the command line tools to verify the fingerprint. For more information about installing the AWS CLI or Amazon EC2 CLI, see Accessing Amazon EC2 (p. 3).

- **Get the ID of the instance**
  You can get the ID of your instance using the Amazon EC2 console (from the **Instance ID** column). If you prefer, you can use the describe-instances (AWS CLI) or ec2-describe-instances (Amazon EC2 CLI) command.

- **Get the public DNS name of the instance**
  You can get the public DNS for your instance using the Amazon EC2 console (check the **Public DNS** column; if this column is hidden, click the **Show/Hide** icon and select **Public DNS**). If you prefer, you can use the describe-instances (AWS CLI) or ec2-describe-instances (Amazon EC2 CLI) command.

- **Locate the private key**
  You'll need the fully-qualified path of the `.pem` file for the key pair that you specified when you launched the instance.

- **Enable inbound SSH traffic from your IP address to your instance**

Ensure that the security group associated with your instance allows incoming SSH traffic from your IP address. For more information, see Authorizing Network Access to Your Instances.

> **Important**
> Your default security group does not allow incoming SSH traffic by default.

Connecting to Your Linux Instance

To connect to your instance using SSH

1. (Optional) You can verify the RSA key fingerprint on your instance by using one of the following commands on your local system (not on the instance). This is useful if you've launched your instance from a public AMI from a third party. Locate the `SSH HOST KEY FINGERPRINTS` section, and note the RSA fingerprint (for example, 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f) and compare it to the fingerprint of the instance.

    *   get-console-output (AWS CLI)

        ```
        aws ec2 get-console-output --instance-id instance_id
        ```

    *   ec2-get-console-output (Amazon EC2 CLI)

        ```
        ec2-get-console-output instance_id
        ```

    > **Note**
    > The `SSH HOST KEY FINGERPRINTS` section is only available after the first boot 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 the name of your private key file is `my-key-pair.pem`, you would use the following command:

    ```
    chmod 400 my-key-pair.pem
    ```

4. Use the **ssh** command to connect to the instance. You'll specify the private key (`.pem`) file and *user_name@public_dns_name*. For Amazon Linux, the user name is `ec2-user`. For RHEL5, the user name is either `root` or `ec2-user`. For Ubuntu, the user name is `ubuntu`. For Fedora, the user name is either `fedora` or `ec2-user`. For SUSE Linux, the user name is `root`. Otherwise, if `ec2-user` and `root` don't work, check with your AMI provider.

    ```
    ssh -i my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com
    ```

    You'll see a response like the following.

    ```
    The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com
    (10.254.142.33)'
    can't be established.
    RSA key fingerprint is
    1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.
    Are you sure you want to continue connecting (yes/no)?
    ```

5. (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you obtained in step 1. If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.

6. Enter `yes`.

    You'll see a response like the following.

    ```
    Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)

    to the list of known hosts.
    ```

Transferring Files to Linux Instances from Linux Using SCP

One way to transfer files between your local computer and a Linux 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

-   **Install an SCP client**
    Most Linux, Unix, and Apple 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.

-   **Get the ID of the instance**
    You can get the ID of your instance using the Amazon EC2 console (from the **Instance ID** column). If you prefer, you can use the describe-instances (AWS CLI) or ec2-describe-instances (Amazon EC2 CLI) command.

-   **Get the public DNS name of the instance**
    You can get the public DNS for your instance using the Amazon EC2 console (check the **Public DNS** column; if this column is hidden, click the **Show/Hide** icon and select **Public DNS**). If you prefer, you can use the describe-instances (AWS CLI) or ec2-describe-instances (Amazon EC2 CLI) command.

-   **Locate the private key**
    You'll need the fully-qualified path of the `.pem` file for the key pair that you specified when you launched the instance.

-   **Enable inbound SSH traffic from your IP address to your instance**
    Ensure that the security group associated with your instance allows incoming SSH traffic from your IP address. For more information, see Authorizing Network Access to Your Instances.

    > **Important**
    > Your default security group does not allow incoming SSH traffic by default.

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

1. (Optional) You can verify the RSA key fingerprint on your instance by using one of the following commands on your local system (not on the instance). This is useful if you've launched your instance from a public AMI from a third party. Locate the `SSH HOST KEY FINGERPRINTS` section, and note the RSA fingerprint (for example, 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f) and compare it to the fingerprint of the instance.

- get-console-output (AWS CLI)

```
aws ec2 get-console-output --instance-id instance_id
```

- ec2-get-console-output (Amazon EC2 CLI)

```
ec2-get-console-output instance_id
```

> **Note**
> The SSH HOST KEY FINGERPRINTS section is only available after the first boot of the instance.

2. In a command shell, change directories to the location of the private key file that you specified when you launched the instance.

3. Use the **chmod** command to make sure your private key file isn't publicly viewable. For example, if the name of your private key file is `my-key-pair.pem`, you would use the following command:

```
chmod 400 my-key-pair.pem
```

4. Transfer a file to your instance using the instance's public DNS name. For example, if the name of the private key file is `my-key-pair`, the file to transfer is `SampleFile.txt`, and the public DNS name of the instance is `ec2-198-51-100-1.compute-1.amazonaws.com`, use the following command to copy the file to the `ec2-user` home directory.

```
scp -i my-key-pair.pem SampleFile.txt ec2-user@ec2-198-51-100-1.compute-
1.amazonaws.com:~
```

> **Tip**
> For Amazon Linux, the user name is `ec2-user`. For RHEL5, the user name is either `root`
> or `ec2-user`. For Ubuntu, the user name is `ubuntu`. For Fedora, the user name is either
> `fedora` or `ec2-user`. For SUSE Linux, the user name is `root`. Otherwise, if `ec2-user`
> and `root` don't work, check with your AMI provider.

You'll see a response like the following.

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com
(10.254.142.33)'
can't be established.
RSA key fingerprint is
1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.
Are you sure you want to continue connecting (yes/no)?
```

5. (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you obtained in step 1. If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.

6. Enter **yes**.

You'll see a response like the following.

```
Warning: Permanently added 'ec2-198-51-100-1.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 (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 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-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com:~/Sample
File.txt ~/SampleFile2.txt
```

Connecting to Your Linux Instance from Windows Using PuTTY

After you launch your instance, you can connect to it and use it the way that you'd use a computer sitting in front of you.

> **Note**
> It may take a couple of minutes after launch for your running instance to finish provisioning so that you can connect to it. Check that your instance has passed its status checks - you can view this information in the **Status Checks** column on the **Instances** page. If you receive an error while attempting to connect to your instance, see Troubleshooting Connecting to Your Instance.

The following instructions explain how to connect to your instance using PuTTY, a free SSH client for Windows.

Prerequisites

- **Install PuTTY**
  Download and install PuTTY from the PuTTY download page. Be sure to install the entire suite.

- **Get the ID of the instance**
  You can get the ID of your instance using the Amazon EC2 console (from the **Instance ID** column). If you prefer, you can use the describe-instances (AWS CLI) or ec2-describe-instances (Amazon EC2 CLI) command.

- **Get the public DNS name of the instance**
  You can get the public DNS for your instance using the Amazon EC2 console (check the **Public DNS** column; if this column is hidden, click the **Show/Hide** icon and select **Public DNS**). If you prefer, you can use the describe-instances (AWS CLI) or ec2-describe-instances (Amazon EC2 CLI) command.

- **Locate the private key**
  You'll need the fully-qualified path of the .pem file for the key pair that you specified when you launched the instance.

- **Enable inbound SSH traffic from your IP address to your instance**
  Ensure that the security group associated with your instance allows incoming SSH traffic from your IP address. For more information, see Authorizing Network Access to Your Instances.

  > **Important**
  > Your default security group does not allow incoming SSH traffic by default.

Converting Your Private Key Using PuTTYgen

PuTTY does not natively support the private key format (.pem) generated by Amazon EC2. PuTTY has a tool named PuTTYgen, which can convert keys to the required PuTTY format (.ppk). You must convert your private key into this format (.ppk) before attempting to connect to your instance using PuTTY.

To convert your private key

1. Start PuTTYgen (for example, from the **Start** menu, click **All Programs > PuTTY > PuTTYgen**).
2. Under **Type of key to generate**, select **SSH-2 RSA**.
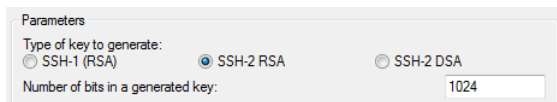
3. Click **Load**. By default, PuTTYgen displays only files with the extension `.ppk`. To locate your `.pem` file, select the option to display files of all types.
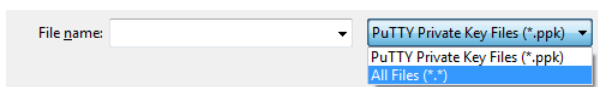
4. Select your `.pem` file and click **Open**. Click **OK** to dismiss the confirmation dialog box.
5. Click **Save private key** to save the key in the format that PuTTY can use. PuTTYgen displays a warning about saving the key without a passphrase. Click **Yes**.

    > **Note**
    > A passphrase on a private key is an extra layer of protection, so even if your private key is discovered, it can't be used 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.

6. Specify the same name for the key that you used for the key pair (for example, `my-key-pair`). 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.

Starting a PuTTY Session

To start a PuTTY session

1. (Optional) You can verify the RSA key fingerprint on your instance by using one of the following commands on your local system (not on the instance). This is useful if you've launched your instance from a public AMI from a third party. Locate the `SSH HOST KEY FINGERPRINTS` section, and note the RSA fingerprint (for example, 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f) and compare it to the fingerprint of the instance.

    -   get-console-output (AWS CLI)

        ```
        aws ec2 get-console-output --instance-id instance_id
        ```

    -   ec2-get-console-output (Amazon EC2 CLI)

```
ec2-get-console-output instance_id
```

> **Note**
>
> The `SSH HOST KEY FINGERPRINTS` section is only available after the first boot of the in-
> stance.

2. Start PuTTY (from the **Start** menu, click **All Programs > PuTTY > PuTTY**).

3. In the Category pane, select **Session** and complete the following fields:

    a. In the **Host Name** box, enter *user_name*@*public_dns_name*. Be sure to specify the appro-
       priate user name for your AMI. For example:

       • For an Amazon Linux AMI, the user name is `ec2-user`.

       • For a RHEL5 AMI, the user name is either `root` or `ec2-user`.

       • For an Ubuntu AMI, the user name is `ubuntu`.

       • For a Fedora AMI, the user name is either `fedora` or `ec2-user`.

       • For SUSE Linux, the user name is `root`.

       • Otherwise, if `ec2-user` and `root` don't work, check with the AMI provider.

    b. Under **Connection type**, select **SSH**.

    c. Ensure that **Port** is 22.
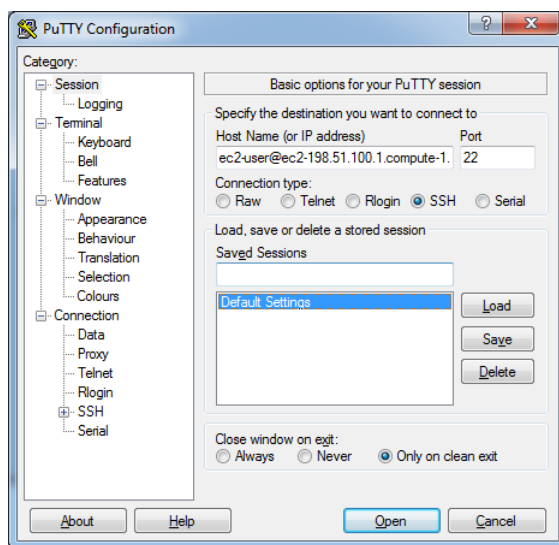
4. In the **Category** pane, expand **Connection**, expand **SSH**, and then select **Auth**. Complete the fol-
   lowing:

    a. Click **Browse**.

    b. Select the `.ppk` file that you generated for your key pair, and then click **Open**.

    c. (Optional) If you plan to start this session again later, you can save the session information for
       future use. Select **Session** in the **Category** tree, enter a name for the session in **Saved Sessions**,
       and then click **Save**.

    d. Click **Open** to start the PuTTY session.
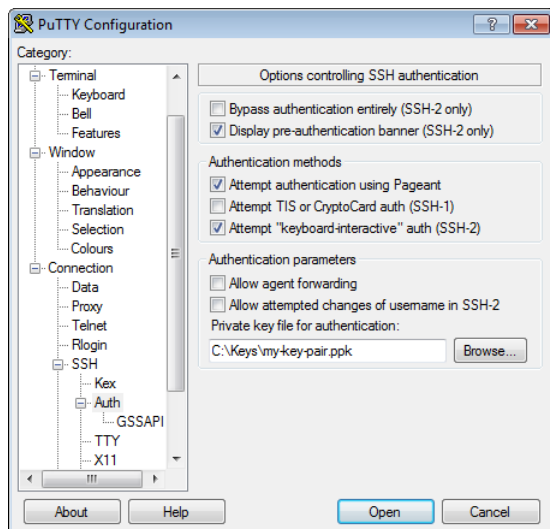
5.   If this is the first time you have connected to this instance, PuTTY displays a security alert dialog box that asks whether you trust the host you are connecting to.

6.   (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you obtained in step 1. If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.

7.   Click **Yes**. A window opens and you are connected to your instance.

> **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.

Transferring Files to Your Linux Instance Using the PuTTY Secure Copy Client

The PuTTY Secure Copy client (PSCP) is a command-line tool that you can use to transfer files between your Windows computer and your Linux instance. If you prefer a graphical user interface (GUI), you can use an open source GUI tool named WinSCP. For more information, see Transferring Files to Your Linux Instance Using WinSCP (p. 306).

To use PSCP, you'll need the private key you generated in Converting Your Private Key Using PuTTY-gen (p. 304). You'll also need the public DNS address of your Linux instance.

The following example transfers the file Sample_file.txt from a Windows computer to the /usr/local directory on a Linux instance:

```
C:\> pscp -i C:\Keys\my-key-pair.ppk C:\Sample_file.txt user_name@pub
lic_dns:/usr/local/Sample_file.txt
```

Transferring Files to Your Linux Instance Using WinSCP

WinSCP is a GUI-based file manager for Windows that allows you to upload and transfer files to a remote computer using the SFTP, SCP, FTP, and FTPS protocols. WinSCP allows you to drag and drop files from your Windows machine to your Linux instance or synchronize entire directory structures between the two systems.

To use WinSCP, you'll need the private key you generated in Converting Your Private Key Using PuTTYgen (p. 304). You'll also need the public DNS address of your Linux instance.

1. Download and install WinSCP from http://winscp.net/eng/download.php. For most users, the default installation options are OK.

2. Start WinSCP.

3. At the **WinSCP login** screen, for **Host name**, enter the public DNS address for your instance.

4. For **User name**, enter the default user name for your AMI. For Amazon Linux AMIs, the user name is `ec2-user`. For Red Hat AMIs the user name is `root`, and for Ubuntu AMIs the user name is `ubuntu`.

5. Specify the private key for your instance. For **Private key**, enter the path to your private key, or click the "…" button to browse for the file. For newer versions of WinSCP, you need to click **Advanced** to open the advanced site settings and then under **SSH**, click **Authentication** to find the **Private key file** setting.

    > **Note**
    > WinSCP requires a PuTTY private key file (`.ppk`). You can convert a `.pem` security key file to the `.ppk` format using PuTTYgen. For more information, see Converting Your Private Key Using PuTTYgen (p. 304).

6. (Optional) In the left panel, click **Directories**, and then, for **Remote directory**, enter the path for the directory you want to add files to. For newer versions of WinSCP, you need to click **Advanced** to open the advanced site settings and then under **Environment**, click **Directories** to find the **Remote directory** setting.

7. Click **Login** to connect, and click **Yes** to add the host fingerprint to the host cache.
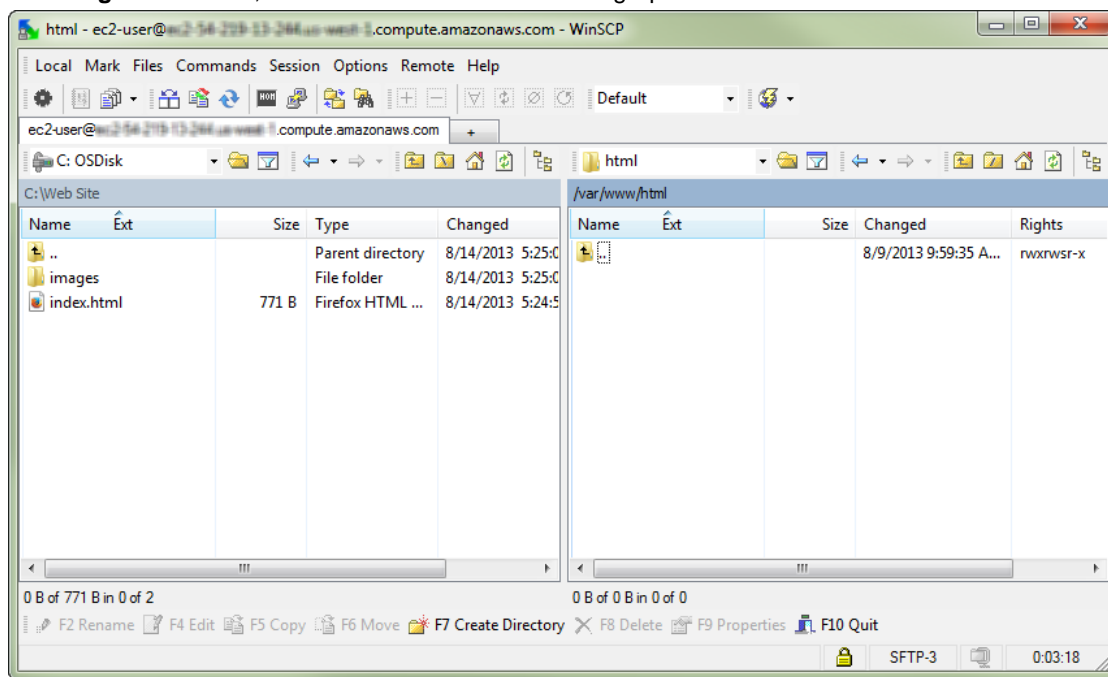
8. After the connection is established, in the connection window your Linux instance is on the right and your local machine is on the left. You can drag and drop files directly into the remote file system from your local machine. For more information on WinSCP, see the project documentation at http://winscp.net/eng/docs/start.

Connecting to Your Linux Instance Using MindTerm

After you launch your instance, you can connect to it and use it the way that you'd use a computer sitting in front of you.

> **Note**
> It may take a couple of minutes after launch for your running instance to finish provisioning so that you can connect to it. Check that your instance has passed its status checks - you can view this information in the **Status Checks** column on the **Instances** page. If you receive an error while attempting to connect to your instance, see Troubleshooting Connecting to Your Instance.

The following instructions explain how to connect to your instance using MindTerm through the Amazon EC2 console.

Prerequisites

- **Install Java**
  Your Linux computer most likely includes Java. If not, see How do I enable Java in my web browser? On a Windows or Mac client, you must run your browser using administrator credentials. For Linux, additional steps may be required if you are not logged in as `root`.

- **Enable Java in your browser**
  For instructions, see http://java.com/en/download/help/enable_browser.xml.

- **Locate the private key**
  You'll need the fully-qualified path of the `.pem` file for the key pair that you specified when you launched the instance.

- **Enable inbound SSH traffic from your IP address to your instance**
  Ensure that the security group associated with your instance allows incoming SSH traffic from your IP address. For more information, see Authorizing Network Access to Your Instances.

  > **Important**
  > Your default security group does not allow incoming SSH traffic by default.

Starting MindTerm

To connect to your instance using a web browser with MindTerm

1. In the Amazon EC2 console, click **Instances** in the navigation pane.
2. Select the instance, and then click **Connect**.
3. Click **A Java SSH client directly from my browser (Java required)**.
4. Amazon EC2 automatically detects the public DNS name of your instance and the name of the populates **Public DNS** for you. It also detects name of the key pair that you specified when you launched the instance. Complete the following, and then click **Launch SSH Client**.

   a. In **User name**, enter the user name to log in to your instance.

      > **Tip**
      > For Amazon Linux, the user name is `ec2-user`. For RHEL5, the user name is either `root` or `ec2-user`. For Ubuntu, the user name is `ubuntu`. For Fedora, the user name is either `fedora` or `ec2-user`. For SUSE Linux, the user name is `root`. Otherwise, if `ec2-user` and `root` don't work, check with your AMI provider.

   b. In **Private key path**, enter the fully-qualified path to your private key (`.pem`) file, including the key pair name; for example:

```
C:\KeyPairs\my-key-pair.pem
```

c. (Optional) Click **Store in browser cache** to store the location of the private key in your browser cache. This enables Amazon EC2 to detect the location of the private key in subsequent browser sessions, until your clear your browser's cache.

5. If necessary, click **Yes** to trust the certificate, and click **Run** to run the MindTerm client.

6. If this is your first time running MindTerm, a series of dialog boxes asks you to accept the license agreement, to confirm setup for your home directory, and to confirm setup of the known hosts directory. Confirm these settings.

7. A dialog prompts you to add the host to your set of known hosts. If you do not want to store the host key information on your local computer, click **No**.

8. A window opens and you are connected to your instance.

> **Note**
> If you clicked **No** in the previous step, you'll see the following message, which is expected:

```
Verification of server key disabled in this session.
```

Stop and Start Your Instance

You can stop and restart your instance if it has an Amazon EBS volume as its root device. The instance retains its instance ID, but can change as described in the Overview section.

When you stop an instance, we shut it down. We don't charge hourly usage for a stopped instance, or data transfer fees, but we do charge for the storage for any Amazon EBS volumes. Each time you start a stopped instance we charge a full instance hour, even if you make this transition multiple times within a single hour.

While the instance is stopped, you can treat its root volume like any other volume, and modify it (for example, repair file system problems or update software). You just detach the volume from the stopped instance, attach it to a running instance, make your changes, detach it from the running instance, and then reattach it to the stopped instance. Make sure that you reattach it using the storage device name that's specified as the root device in the block device mapping for the instance.

If you decide that you no longer need an instance, you can terminate it. As soon as the state of an instance changes to `shutting-down` or `terminated`, we stop charging for that instance. For more information, see Terminate Your Instance (p. 315).

Topics

Overview

You can only stop an Amazon EBS-backed instance. To verify the root device type of your instance, describe the instance and check whether the device type of its root volume is `ebs` (Amazon EBS-backed instance) or `instance store` (instance store-backed instance). For more information, see Determining the Root Device Type of Your AMI (p. 56).

When you stop a running instance, the following happens:

- The instance performs a normal shutdown and stops running; its status changes to `stopping` and then `stopped`.
- Any Amazon EBS volumes remain attached to the instance, and their data persists.
- Any data stored in the RAM of the host computer or the instance store volumes of the host computer is gone.
- EC2-Classic: We release the public and private IP addresses for the instance when you stop the instance, and assign new ones when you restart it.

  EC2-VPC: The instance retains its private IP addresses when stopped and restarted. We release the public IP address and assign a new one when you restart it.
- EC2-Classic: We disassociate any Elastic IP address (EIP) that's associated with the instance. You're charged for Elastic IP addresses that aren't associated with an instance. When you restart the instance, you must associate the Elastic IP address with the instance; we don't do this automatically.

  EC2-VPC: The instance retains its associated Elastic IP addresses (EIP). You're charged for any Elastic IP addresses associated with a stopped instance.
- If you've registered the instance with a load balancer, it's likely that the load balancer won't be able to route traffic to your instance after you've stopped and restarted it. You must de-register the instance from the load balancer after stopping the instance, and then re-register after starting the instance. For more information, see De-Registering and Registering Amazon EC2 Instances in the *Elastic Load Balancing Developer Guide*.

For more information, see Differences Between Reboot, Stop, and Terminate (p. 288).

You can modify the following attributes of an instance only when it is stopped:

- Instance type
- User data
- Kernel
- RAM disk

If you try to modify these attributes while the instance is running, Amazon EC2 returns the `Incorrect-InstanceState` error.

Stopping and Starting Your Instances

You can start and stop your Amazon EBS-backed instance using the console or the command line.

By default, when you initiate a shutdown from an Amazon EBS-backed instance (using the **shutdown**, **halt**, or **poweroff** command), the instance stops. You can change this behavior so that it terminates instead. For more information, see Changing the Instance Initiated Shutdown Behavior (p. 317).

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

1. In the navigation pane, click **Instances**, and select the instance.
2. [EC2-Classic] If the instance has an associated Elastic IP address, write down the Elastic IP address and the instance ID shown in the details pane.
3. Click **Actions**, and then click **Stop**. If **Stop** is disabled, either the instance is already stopped or its root device is an instance store volume.
4. In the confirmation dialog box, click **Yes, Stop**. It can take a few minutes for the instance to stop.

[EC2-Classic] When the instance state becomes `stopped`, the **Elastic IP**, **Public DNS**, **Private DNS**, and **Private IPs** fields in the details pane are blank to indicate that the old values are no longer associated with the instance.

5. While your instance is stopped, you can modify certain instance attributes. For more information, see Modifying a Stopped Instance (p. 311).

6. To restart the stopped instance, select the instance, click **Actions**, and then click **Start**.

7. In the confirmation dialog box, click **Yes, Start**. It can take a few minutes for the instance to enter the `running` state.

    [EC2-Classic] When the instance state becomes `running`, the **Public DNS**, **Private DNS**, and **Private IPs** fields in the details pane contain the new values that we assigned to the instance.

8. [EC2-Classic] If your instance had an associated Elastic IP address, you must reassociate it as follows:

    a. In the navigation pane, click **Elastic IPs**.
    b. Select the Elastic IP address that you wrote down before you stopped the instance.
    c. Click **Associate Address**.
    d. Select the instance ID that you wrote down before you stopped the instance, and then click **Associate**.

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

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

-   stop-instances and start-instances (AWS CLI)
-   ec2-stop-instances and ec2-start-instances (Amazon EC2 CLI)
-   Stop-EC2Instance and Start-EC2Instance (AWS Tools for Windows PowerShell)

Modifying a Stopped Instance

You can change the instance type and user data attributes using the AWS Management Console or the command line interface. You can't use the AWS Management Console to modify the kernel or RAM disk attributes.

To change the instance type for a stopped instance using the console

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

    For more information, see Resizing Your Instance (p. 128).

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

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

To modify an instance attribute using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- modify-instance-attribute (AWS CLI)
- ec2-modify-instance-attribute (Amazon EC2 CLI)
- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

Troubleshooting

If you have stopped your Amazon EBS-backed instance and it appears "stuck" in the `stopping` state, you can forcibly stop it. For more information, see Troubleshooting Stopping Your Instance (p. 637).

Reboot Your Instance

An instance reboot is equivalent to an operating system reboot. In most cases, it takes only a few minutes to reboot your instance. When you reboot an instance, it remains on the same physical host, so your instance keeps its public DNS name, private IP address, and any data on its instance store volumes.

Rebooting an instance doesn't start a new instance billing hour, unlike stopping and restarting your instance.

We might schedule your instance for a reboot for necessary maintenance, such as to apply updates that require a reboot. No action is required on your part; we recommend that you wait for the reboot to occur within its scheduled window. For more information, see Monitoring Events for Your Instances (p. 351).

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

To reboot an instance using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**.
3. Select the instance, click **Actions**, and then click **Reboot**.
4. Click **Yes, Reboot** when prompted for confirmation.

To reboot an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- reboot-instances (AWS CLI)
- ec2-reboot-instances (Amazon EC2 CLI)
- Restart-EC2Instance (AWS Tools for Windows PowerShell)

Instance Retirement

An instance is scheduled to be retired when AWS detects irreparable failure of the underlying hardware hosting the instance. When an instance reaches its scheduled retirement date, it is stopped or terminated by AWS. If your instance root device is an Amazon EBS volume, the instance is stopped, and you can

start it again at any time. Starting the stopped instance migrates it to new hardware. If your instance root device is an instance store volume, the instance is terminated, and cannot be used again.

Topics
- Identifying Instances Scheduled for Retirement (p. 313)
- Working with Instances Scheduled for Retirement (p. 313)

For more information about types of instance events, see Monitoring Events for Your Instances (p. 351).

Identifying Instances Scheduled for Retirement

If your instance is scheduled for retirement, you'll receive an email prior to the event with the instance ID and retirement date. This email is sent to the address that's associated with your account; the same email address that you use to log in to the AWS Management Console. If you use an email account that you do not check regularly, then you can use the Amazon EC2 console or the command line to determine if any of your instances are scheduled for retirement. To update the contact information for your account, go to the Account Settings page.

To identify instances scheduled for retirement using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **EC2 Dashboard**. Under **Scheduled Events**, you can see the events associated with your Amazon EC2 instances and volumes, organized by region.

Scheduled Events

US East (N. Virginia):
1 instances have scheduled events
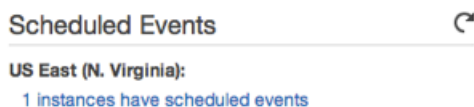
3. If you have an instance with a scheduled event listed, click its link below the region name to go to the **Events** page.
4. The **Events** page lists all resources with events associated with them. To view instances that are scheduled for retirement, select **Instance resources** from the first filter list, and then **Instance retirement** from the second filter list.
5. If the filter results show that an instance is scheduled for retirement, select it, and note the date and time in the **Start time** field in the details pane. This is your instance retirement date.

To identify instances scheduled for retirement using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-instance-status (AWS CLI)
- ec2-describe-instance-status (Amazon EC2 CLI)
- Get-EC2InstanceStatus (AWS Tools for Windows PowerShell)

Working with Instances Scheduled for Retirement

There are a number of actions available to you when your instance is scheduled for retirement. The action you take depends on whether your instance root device is an Amazon EBS volume, or an instance store volume. If you do not know what your instance root device type is, you can find out using the Amazon EC2 console or the command line.

Determining Your Instance Root Device Type

To determine your instance root device type using the console

1. In the navigation pane, click **Events**. Use the filter lists to identify retiring instances, as demonstrated in the procedure above, Identifying instances scheduled for retirement (p. 313).
2. In the **Resource ID** column, click the instance ID to go to the **Instances** page.
3. Select the instance and locate the **Root device type** field in the **Description** tab. If the value is `ebs`, then your instance is EBS-backed. If the value is `instance-store`, then your instance is instance store-backed.

To determine your instance root device type using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-instances (AWS CLI)
- ec2-describe-instances (Amazon EC2 CLI)
- Get-EC2Instance (AWS Tools for Windows PowerShell)

Managing Instances Scheduled for Retirement

You can perform one of the actions listed below in order to preserve the data on your retiring instance. It's important that you take this action before the instance retirement date, to prevent unforeseen downtime and data loss.

> **Warning**
> If your instance store-backed instance passes its retirement date, it's terminated and you cannot recover the instance or any data that was stored on it. Regardless of the root device of your instance, the data on instance store volumes is lost when the instance is retired, even if they are attached to an EBS-backed instance.

Instance Root Device Type	Action
EBS	Wait for the scheduled retirement date - when the instance is stopped - or stop the instance yourself before the retirement date. You can start the instance again at any time. For more information about stopping and starting your instance, and what to expect when your instance is stopped, such as the effect on public, private and Elastic IP addresses associated with your instance, see Stop and Start Your Instance (p. 309).
EBS	Create an EBS-backed AMI from your instance, and launch a replacement instance. For more information, see Creating an Amazon EBS-Backed Linux AMI (p. 74).
Instance store	Create an instance store-backed AMI from your instance using the AMI tools, and launch a replacement instance. For more information, see Creating an Instance Store-Backed Linux AMI (p. 77).
Instance store	Convert your instance to an EBS-backed instance by transferring your data to an EBS volume, taking a snapshot of the volume, and then creating an AMI from the snapshot. You can launch a replacement instance from your new AMI. For more information, see Converting your Instance Store-Backed AMI to an Amazon EBS-Backed AMI (p. 82).

Terminate Your Instance

When you've decided that you no longer need an instance, you can terminate it. As soon as the state of an instance changes to `shutting-down` or `terminated`, you stop incurring charges for that instance.

You can't connect to or restart an instance after you've terminated it. However, you can launch additional instances using the same AMI. If you'd rather stop and restart your instance, see Stop and Start Your Instance (p. 309). For more information, see Differences Between Reboot, Stop, and Terminate (p. 288).

Topics

- Instance Termination (p. 315)
- Terminating an Instance (p. 315)
- Enabling Termination Protection for an Instance (p. 316)
- Changing the Instance Initiated Shutdown Behavior (p. 317)
- Preserving Amazon EBS Volumes on Instance Termination (p. 318)
- Troubleshooting (p. 320)

Instance Termination

After you terminate an instance, it remains visible in the console for a short while, and then the entry is deleted.

When an instance terminates, the data on any instance store volumes associated with that instance is deleted.

By default, any Amazon EBS volumes that you attach as you launch the instance are automatically deleted when the instance terminates. However, by default, any volumes that you attach to a running instance persist even after the instance terminates. This behavior is controlled by the volume's `DeleteOnTermination` attribute, which you can modify. For more information, see Preserving Amazon EBS Volumes on Instance Termination (p. 318).

You can prevent an instance from being terminated accidentally by someone using the AWS Management Console, the CLI, and the API. This feature is available for both Amazon EC2 instance store-backed and Amazon EBS-backed instances. Each instance has a `DisableApiTermination` attribute with the default value of `false` (the instance can be terminated through Amazon EC2). You can modify this instance attribute while the instance is running or stopped (in the case of Amazon EBS-backed instances). For more information, see Enabling Termination Protection for an Instance (p. 316).

You can control whether an instance should stop or terminate when shutdown is initiated from the instance using an operating system command for system shutdown. For more information, see Changing the Instance Initiated Shutdown Behavior (p. 317).

If you run a script on instance termination, your instance might have an abnormal termination, because we have no way to ensure that shutdown scripts run. Amazon EC2 attempts to shut an instance down cleanly and run any system shutdown scripts; however, certain events (such as hardware failure) may prevent these system shutdown scripts from running.

Terminating an Instance

You can terminate an instance using the AWS Management Console or the command line.

To terminate an instance using the console

1.   Before you terminate the instance, verify that you won't lose any data by checking that your Amazon EBS volumes won't be deleted on termination and that you've copied any data that you need from your instance store volumes to Amazon EBS or Amazon S3.

2.   Open the Amazon EC2 console.

3.   In the navigation pane, click **Instances**.

4.   Select the instance, click **Actions**, and then click **Terminate**.

5.   Click **Yes, Terminate** when prompted for confirmation.

To terminate an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

*   terminate-instances (AWS CLI)
*   ec2-terminate-instances (Amazon EC2 CLI)
*   Stop-EC2Instance (AWS Tools for Windows PowerShell)

Enabling Termination Protection for an Instance

By default, you can terminate your instance using the Amazon EC2 console, command line interface, or API. If you want to prevent your instance from being accidentally terminated using Amazon EC2, you can enable *termination protection* for the instance. The `DisableApiTermination` attribute controls whether the instance can be terminated using the console, CLI, or API. By default, termination protection is disabled for your instance. You can set the value of this attribute when you launch the instance, while the instance is running, or while the instance is stopped (for Amazon EBS-backed instances).

The `DisableApiTermination` attribute does not prevent you from terminating an instance by initiating shutdown from the instance (using an operating system command for system shutdown) when the `InstanceInitiatedShutdownBehavior` attribute is set. For more information, see Changing the Instance Initiated Shutdown Behavior (p. 317).

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

You can enable or disable termination protection using the AWS Management Console or the command line.

To enable termination protection for an instance at launch time

1.   On the dashboard of the Amazon EC2 console, click **Launch Instance** and follow the directions in the wizard.

2.   On the **Configure Instance Details** page, select the **Enable termination protection** check box.

To enable termination protection for a running or stopped instance

1.   Select the instance, click **Actions**, and then click **Change Termination Protection**.

2.   Click **Yes, Enable**.

To disable termination protection for a running or stopped instance

1.   Select the instance, click **Actions**, and then click **Change Termination Protection**.

2.   Click **Yes, Disable**.

To enable or disable termination protection using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- modify-instance-attribute (AWS CLI)
- ec2-modify-instance-attribute (Amazon EC2 CLI)
- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

Changing the Instance Initiated Shutdown Behavior

By default, when you initiate a shutdown from an Amazon EBS-backed instance (using a command such as **shutdown**, **halt**, or **poweroff**), the instance stops. You can change this behavior using the `InstanceInitiatedShutdownBehavior` attribute for the instance so that it terminates instead. You can update this attribute while the instance is running or stopped.

You can update the `InstanceInitiatedShutdownBehavior` attribute using the AWS Management Console or the command line.

To change the shutdown behavior of an instance using the console

1.   Open the Amazon EC2 console.

2.   In the navigation pane, click **Instances**.

3.   Select the instance, click **Actions**, and then click **Change Shutdown Behavior**. The current behavior is already selected.

4.   To change the behavior, select an option from the **Shutdown behavior** list, and then click **Apply**.
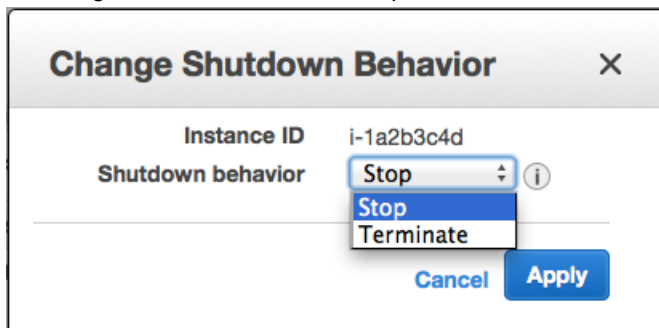
To change the shutdown behavior of an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- modify-instance-attribute (AWS CLI)
- ec2-modify-instance-attribute (Amazon EC2 CLI)
- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

Preserving Amazon EBS Volumes on Instance Termination

By default, we do the following:

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

You can change this behavior using the `DeleteOnTermination` attribute for the volume. If the value of this attribute is `true`, we delete the volume after the instance terminates; otherwise, we preserve the volume. If the `DeleteOnTermination` attribute of a volume is `false`, the volume persists in its current state. You can take a snapshot of the volume, and you can attach it to another instance.

If you detach a volume that you attached to your instance at launch, and then reattach it, we preserve it even after the instance terminates. In other words, its `DeleteOnTermination` attribute is set to `false`.

You can see the value for the `DeleteOnTermination` attribute on the volumes attached to an instance by looking at the instance's block device mapping. For more information, see Viewing the Amazon EBS Volumes in an Instance Block Device Mapping (p. 600).

You can update the `DeleteOnTermination` attribute using the AWS Management Console or the command line.

Changing the Root Volume to Persist Using the Console

Using the console, you can change the `DeleteOnTermination` attribute when you launch an instance. To change this attribute for a running instance, you must use the command line.

To change the root volume of an instance to persist at launch using the console

1. Open the Amazon EC2 console.
2. From the console dashboard, click **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, choose an AMI and click **Select**.
4. Follow the wizard to complete the **Choose an Instance Type** and **Configure Instance Details** pages.
5. On the **Add Storage** page, deselect the **Delete On Termination** check box for the root volume.
6. Complete the remaining wizard pages, and then click **Launch**.

You can verify the setting by viewing details for the root device volume on the instance's details pane. Next to **Block devices**, click the entry for the root device volume. By default, **Delete on termination** is `True`. If you change the default behavior, **Delete on termination** is `False`.

Changing the Root Volume of a Running Instance to Persist Using the Command Line

You can use one of the following commands to change the root device volume of a running instance to persist. The root device is typically `/dev/sda1`. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- modify-instance-attribute (AWS CLI)

- ec2-modify-instance-attribute (Amazon EC2 CLI)
- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

Example for AWS CLI

The following command preserves the root volume by setting its `DeleteOnTermination` attributes to
`false`.

```
$ aws ec2 modify-instance-attribute --instance-id i-5203422c --block-device-
mappings  '[{"DeviceName":"/dev/sda1","Ebs":{"DeleteOnTermination":false}}]'
```

You can confirm that `deleteOnTermination` is `false` by using the describe-instances command and
looking for the `BlockDeviceMappings` entry for `/dev/sda1` in the command output.

Example for Amazon EC2 CLI

The following command preserves the root volume by setting its `DeleteOnTermination` attribute to
`false`.

```
$ ec2-modify-instance-attribute i-5203422c -b /dev/sda1=::false
```

Changing the Root Volume of an Instance to Persist at Launch Using the Command Line

When you launch an instance, you can use one of the following commands to change the root device
volume to persist. The root device is typically `/dev/sda1`. For more information about these command
line interfaces, see Accessing Amazon EC2 (p. 3).

- run-instances (AWS CLI)
- ec2-run-instances (Amazon EC2 CLI)
- New-EC2Instance (AWS Tools for Windows PowerShell)

Example for AWS CLI

The following command preserves the root volume by setting its `DeleteOnTermination` attributes to
`false`.

```
$ aws ec2 run-instances --image-id ami-1a2b3c4d --block-device-mappings
'[{"DeviceName":"/dev/sda1","Ebs":{"DeleteOnTermination":false}}]" other para
meters...
```

You can confirm that `deleteOnTermination` is `false` by using the describe-instances command and
looking for the `BlockDeviceMappings` entry for `/dev/sda1` in the command output.

Example for Amazon EC2 CLI

The following command preserves the root volume by setting its `DeleteOnTermination` attribute to
`false`.

```
$ ec2-run-instances ami-1a2b3c4d  -b /dev/sda1=::false other parameters... -v
```

Troubleshooting

If your instance is in the `shutting-down` state for longer than usual, it will eventually be cleaned up (terminated) by automated processes within the Amazon EC2 service. For more information, see Troubleshooting Terminating (Shutting Down) Your Instance (p. 638).

Configuring Your Amazon Linux Instance

After you have successfully launched and logged into your Amazon Linux instance, you can make changes to it. There are many different ways you can configure an instance to meet the needs of a specific application. The following are some common tasks to help get you started.

Topics

Common Configuration Scenarios

The base distribution of Amazon Linux contains many software packages and utilities that are required for basic server operations. However, many more software packages are available in various software repositories, and even more packages are available for you to build from source code. For more information on installing and building software from these locations, see Managing Software on Your Linux Instance (p. 322).

Amazon Linux instances come pre-configured with an `ec2-user` account, but you may want to add other user accounts that do not have super-user privileges. For more information on adding and removing user accounts, see Managing User Accounts on Your Linux Instance (p. 330).

The default time configuration for Amazon Linux instances uses Network Time Protocol to set the system time on an instance. The default time zone is UTC. For more information on setting the time zone for an instance or using your own time server, see Setting the Time for Your Linux Instance (p. 332).

If you have your own network with a domain name registered to it, you can change the hostname of an instance to identify itself as part of that domain. You can also change the system prompt to show a more meaningful name without changing the hostname settings. For more information, see Changing the

You can configure an instance to use a dynamic DNS service provider. For more information, see .

When you launch an instance in Amazon EC2, you have the option of passing user data to the instance that can be used to perform common configuration tasks and even run scripts after the instance starts. You can pass two types of user data to Amazon EC2, `cloud-init` directives, and shell scripts. For more information, see .

Managing Software on Your Linux Instance

The base distribution of Amazon Linux contains many software packages and utilities that are required for basic server operations. However, many more software packages are available in various software repositories, and even more packages are available for you to build from source code.

Contents

It is important to keep software up-to-date. Many packages in a Linux distribution are updated frequently to fix bugs, add features, and protect against security exploits. For more information, see Updating Instance Software (p. 322).

By default, Amazon Linux instances launch with two repositories enabled: `amzn-main` and `amzn-updates`. While there are many packages available in these repositories that are updated by Amazon Web Services, there may be a package that you wish to install that is contained in another repository. For more information, see Adding Repositories (p. 326). For help finding packages in enabled repositories, see Finding Software Packages (p. 327). For information about installing software on an Amazon Linux instance, see Installing Software Packages (p. 328).

Not all software is available in software packages stored in repositories; some software must be compiled on an instance from its source code. For more information, see Preparing to Compile Software (p. 329).

Amazon Linux instances manage their software using the yum package manager. The yum package manager can install, remove, and update software, as well as manage all of the dependencies for each package. Debian-based Linux distributions, like Ubuntu, use the **apt-get** command and **dpkg** package manager, so the **yum** examples in the following sections do not work for those distributions.

Updating Instance Software

It is important to keep software up-to-date. Many packages in a Linux distribution are updated frequently to fix bugs, add features, and protect against security exploits. When you first launch and connect to an Amazon Linux instance, you may see a message asking you to update software packages for security purposes. This section shows how to update an entire system, or just a single package.

Important
These procedures are intended for use with Amazon Linux, but the commands and file locations are similar for Red Hat and CentOS. For more information about other distributions, see their specific documentation.

```
    __|  __|_  )
    _|  (     /    Amazon Linux AMI
   ___|\___|___|
https://aws.amazon.com/amazon-linux-ami/2013.03-release-notes/
There are 12 security update(s) out of 25 total update(s) available
Run "sudo yum update" to apply all updates.
[ec2-user ~]$
```

To update all packages on an Amazon Linux instance

1. (Optional) Start a **screen** session in your shell window. Sometimes you may experience a network interruption that can disconnect the SSH connection to your instance. If this happens during a long software update, it can leave the instance in a recoverable, although confused state. A **screen** session allows you to continue running the update even if your connection is interrupted, and you can reconnect to the session later without problems.

    a. Execute the **screen** command to begin the session.

    ```
    [ec2-user ~]$ screen
    ```

    b. If your session is disconnected, log back into your instance and list the available screens.

    ```
    [ec2-user ~]$ screen -ls
    There is a screen on:
     17793.pts-0.ip-12-34-56-78 (Detached)
    1 Socket in /var/run/screen/S-ec2-user.
    ```

    c. Reconnect to the screen using the **screen -r** command and the process ID from the previous command.

    ```
    [ec2-user ~]$ screen -r 17793
    ```

    d. When you are finished using **screen**, use the **exit** command to close the session.

    ```
    [ec2-user ~]$ exit
    [screen is terminating]
    ```

2. Run the **yum update** command. Optionally, you can add the `--security` flag to apply only security updates.

    ```
    [ec2-user ~]$ sudo yum update
    Loaded plugins: priorities, security, update-motd, upgrade-helper
    amzn-main                                             | 2.1 kB     00:00
    amzn-updates                                          | 2.3 kB     00:00
    Setting up Update Process
    Resolving Dependencies
    --> Running transaction check
    ---> Package aws-apitools-ec2.noarch 0:1.6.8.1-1.0.amzn1 will be updated
    ---> Package aws-apitools-ec2.noarch 0:1.6.10.0-1.0.amzn1 will be an update
    ```

```
---> Package gnupg2.x86_64 0:2.0.18-1.16.amzn1 will be updated
---> Package gnupg2.x86_64 0:2.0.19-8.21.amzn1 will be an update
---> Package libgcrypt.i686 0:1.4.5-9.10.amzn1 will be updated
---> Package libgcrypt.x86_64 0:1.4.5-9.10.amzn1 will be updated
---> Package libgcrypt.i686 0:1.4.5-9.12.amzn1 will be an update
---> Package libgcrypt.x86_64 0:1.4.5-9.12.amzn1 will be an update
---> Package openssl.x86_64 1:1.0.1e-4.53.amzn1 will be updated
---> Package openssl.x86_64 1:1.0.1e-4.54.amzn1 will be an update
---> Package python-boto.noarch 0:2.9.9-1.0.amzn1 will be updated
---> Package python-boto.noarch 0:2.13.3-1.0.amzn1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================

 Package              Arch        Version                    Repository
 Size
================================================================================
Updating:
 aws-apitools-ec2     noarch      1.6.10.0-1.0.amzn1         amzn-updates
 14 M
 gnupg2               x86_64      2.0.19-8.21.amzn1          amzn-updates
2.4 M
 libgcrypt            i686        1.4.5-9.12.amzn1           amzn-updates
248 k
 libgcrypt            x86_64      1.4.5-9.12.amzn1           amzn-updates
262 k
 openssl              x86_64      1:1.0.1e-4.54.amzn1        amzn-updates
1.7 M
 python-boto          noarch      2.13.3-1.0.amzn1           amzn-updates
1.6 M

Transaction Summary
================================================================================
Upgrade       6 Package(s)

Total download size: 20 M
Is this ok [y/N]:
```

3. Review the packages listed, and type **y** and **Enter** to accept the updates. Updating all of the packages on a system can take several minutes. The **yum** output shows the status of the update while it is running.

```
Downloading Packages:
(1/6): aws-apitools-ec2-1.6.10.0-1.0.amzn1.noarch.rpm    |  14 MB      00:00
(2/6): gnupg2-2.0.19-8.21.amzn1.x86_64.rpm               | 2.4 MB      00:00
(3/6): libgcrypt-1.4.5-9.12.amzn1.i686.rpm               | 248 kB      00:00
(4/6): libgcrypt-1.4.5-9.12.amzn1.x86_64.rpm             | 262 kB      00:00
(5/6): openssl-1.0.1e-4.54.amzn1.x86_64.rpm              | 1.7 MB      00:00
(6/6): python-boto-2.13.3-1.0.amzn1.noarch.rpm           | 1.6 MB      00:00
--------------------------------------------------------------------------
------
Total                                          28 MB/s |  20 MB      00:00
Running rpm_check_debug
Running Transaction Test
```

```
Transaction Test Succeeded
Running Transaction
  Updating    : libgcrypt-1.4.5-9.12.amzn1.x86_64
 1/12
  Updating    : gnupg2-2.0.19-8.21.amzn1.x86_64
 2/12
  Updating    : aws-apitools-ec2-1.6.10.0-1.0.amzn1.noarch
 3/12
  Updating    : 1:openssl-1.0.1e-4.54.amzn1.x86_64
 4/12
  ...

Complete!
```

To update a single package on an Amazon Linux instance

Use this procedure to update a single package (and its dependencies) and not the entire system.

1. Run the **yum update** command with the name of the package you would like to update.

```
[ec2-user ~]$ sudo yum update openssl
Loaded plugins: priorities, security, update-motd, upgrade-helper
amzn-main                                                      | 2.1 kB     00:00
amzn-updates                                                   | 2.3 kB     00:00
Setting up Update Process
Resolving Dependencies
--> Running transaction check
---> Package openssl.x86_64 1:1.0.1e-4.53.amzn1 will be updated
---> Package openssl.x86_64 1:1.0.1e-4.54.amzn1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved


================================================================================

 Package         Arch         Version                       Repository
 Size
================================================================================
Updating:
 openssl         x86_64       1:1.0.1e-4.54.amzn1           amzn-updates
1.7 M

Transaction Summary
================================================================================
Upgrade         1 Package(s)

Total download size: 1.7 M
Is this ok [y/N]:
```

2. Review the package information listed, and type **y** and **Enter** to accept the update or updates.
    Sometimes there will be more than one package listed if there are package dependencies that must
    be resolved. The **yum** output shows the status of the update while it is running.

```
Downloading Packages:
openssl-1.0.1e-4.54.amzn1.x86_64.rpm                        | 1.7 MB     00:00
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating   : 1:openssl-1.0.1e-4.54.amzn1.x86_64
  1/2
  Cleanup    : 1:openssl-1.0.1e-4.53.amzn1.x86_64
  2/2
  Verifying  : 1:openssl-1.0.1e-4.54.amzn1.x86_64
  1/2
  Verifying  : 1:openssl-1.0.1e-4.53.amzn1.x86_64
  2/2

Updated:
  openssl.x86_64 1:1.0.1e-4.54.amzn1

Complete!
```

Adding Repositories

By default, Amazon Linux instances launch with two repositories enabled: amzn-main and amzn-updates. While there are many packages available in these repositories that are updated by Amazon Web Services, there may be a package that you wish to install that is contained in another repository.

Important
These procedures are intended for use with Amazon Linux, but the commands and file locations are similar for Red Hat and CentOS. For more information about other distributions, see their specific documentation.

To install a package from a different repository with **yum**, you need to add the repository information to the /etc/yum.conf file or to its own *repository*.repo file in the /etc/yum.repos.d directory. You can do this manually, but most yum repositories provide their own *repository*.repo file at their repository URL.

To add a yum repository to /etc/yum.repos.d

1. Find the location of the .repo file. This will vary depending on the repository you are adding. In this example, the .repo file is at https://www.*example*.com/*repository*.repo.

2. Add the repository with the **yum-config-manager** command.

```
[ec2-user ~]$ sudo yum-config-manager --add-repo https://www.example.com/re
pository.repo
Loaded plugins: priorities, update-motd, upgrade-helper
adding repo from: https://www.example.com/repository.repo
grabbing file https://www.example.com/repository.repo to /etc/yum.repos.d/re
pository.repo
repository.repo                                            | 4.0 kB     00:00
repo saved to /etc/yum.repos.d/repository.repo
```

To enable a yum repository in `/etc/yum.repos.d`

- Use the **yum-config-manager** command with the `--enable` *repository* flag. The following command enables the Extra Packages for Enterprise Linux (EPEL) repository from the Fedora project. By default, this repository is present in `/etc/yum.repos.d` on Amazon Linux instances, but it is not enabled.

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

Note
For information on enabling the EPEL repository on other distributions, such as Red Hat and CentOS, see the EPEL documentation at https://fedoraproject.org/wiki/EPEL.

Finding Software Packages

You can use the **yum search** command to search the descriptions of packages that are available in your configured repositories. This is especially helpful if you don't know the exact name of the package you want to install. Simply append the keyword search to the command; for multiple word searches, wrap the search query with quotation marks.

Important
These procedures are intended for use with Amazon Linux, but the commands and file locations are similar for Red Hat and CentOS. For more information about other distributions, see their specific documentation.

Multiple word search queries in quotation marks only return results that match the exact query. If you don't see the expected package, simplify your search to one keyword and then scan the results. You can also try keyword synonyms to broaden your search.

```
[ec2-user ~]$ sudo yum search "find"
Loaded plugins: priorities, security, update-motd, upgrade-helper
============================= N/S Matched: find ================================
findutils.x86_64 : The GNU versions of find utilities (find and xargs)
perl-File-Find-Rule.noarch : Perl module implementing an alternative interface

                            : to File::Find
perl-Module-Find.noarch : Find and use installed modules in a (sub)category
libpuzzle.i686 : Library to quickly find visually similar images (gif, png,
jpg)
libpuzzle.x86_64 : Library to quickly find visually similar images (gif, png,
                   : jpg)
mlocate.x86_64 : An utility for finding files by name
```

The yum package manager also combines several packages into groups that you can install with one command to perform a particular task, such as installing a web server or build tools for software compilation. To list the groups that are already installed on your system and the available groups that you can install, use the **yum grouplist** command.

```
[ec2-user ~]$ sudo yum grouplist
Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Group Process
Installed Groups:
   Development Libraries
   Development tools
```

```
   Editors
   Legacy UNIX compatibility
   Mail Server
   MySQL Database
   Network Servers
   Networking Tools
   PHP Support
   Perl Support
   System Tools
   Web Server
Available Groups:
   Console internet tools
   DNS Name Server
   FTP Server
   Java Development
   MySQL Database client
   NFS file server
   Performance Tools
   PostgreSQL Database client (version 8)
   PostgreSQL Database server (version 8)
   Scientific support
   TeX support
   Technical Writing
   Web Servlet Engine
Done
```

You can see the different packages in a group by using the **yum groupinfo "*Group Name*"** command, replacing *Group Name* with the name of the group to get information about. This command lists all of the mandatory, default, and optional packages that can be installed with that group.

Installing Software Packages

The yum package manager is a great tool for installing software, because it can search all of your enabled repositories for different software packages and also handle any dependencies in the software installation process.

> **Important**
> These procedures are intended for use with Amazon Linux, but the commands and file locations are similar for Red Hat and CentOS. For more information about other distributions, see their specific documentation.

To install a package from a repository, use the **yum install *package*** command, replacing *package* with the name of the software to install. For example, to install the **links** text-based web browser, enter the following command.

```
[ec2-user ~]$ sudo yum install links
```

To install a group of packages, use the **yum groupinstall *Group Name*** command, replacing *Group Name* with the name of the group you would like to install. For example, to install the "Performance Tools" group, enter the following command.

```
[ec2-user@ip-10-161-113-54 ~]$ sudo yum groupinstall "Performance Tools"
```

By default, yum will only install the mandatory and default packages in the group listing. If you would like to install the optional packages in the group also, you can set the group_package_types configuration parameter in the command when you execute it that adds the optional packages.

```
[ec2-user ~]$ sudo yum --setopt=group_package_types=mandatory,default,optional
 groupinstall "Performance Tools"
```

You can also use **yum install** to install RPM package files that you have downloaded from the Internet.
To do this, simply append the path name of an RPM file to the installation command instead of a repository
package name.

```
[ec2-user ~]$ sudo yum install my-package.rpm
```

Preparing to Compile Software

There is a wealth of open-source software available on the Internet that has not been pre-compiled and
made available for download from a package repository. You may eventually discover a software package
that you need to compile yourself, from its source code. For your system to be able to compile software,
you need to install several development tools, such as **make**, **gcc**, and **autoconf**.

> **Important**
> These procedures are intended for use with Amazon Linux, but the commands and file locations
> are similar for Red Hat and CentOS. For more information about other distributions, see their
> specific documentation.

Because software compilation is not a task that every Amazon EC2 instance requires, these tools are
not installed by default, but they are available in a package group called "Development Tools" that is
easily added to an instance with the **yum groupinstall** command.

```
[ec2-user ~]$ sudo yum groupinstall "Development Tools"
```

Software source code packages are often available for download (from web sites such as https://github.com/
and http://sourceforge.net/) as a compressed archive file, called a tarball. These tarballs will usually have
the `.tar.gz` file extension. You can decompress these archives with the **tar** command.

```
[ec2-user ~]$ tar -xzf software.tar.gz
```

After you have decompressed and unarchived the source code package, you should look for a `README`
or `INSTALL` file in the source code directory that can provide you with further instructions for compiling
and installing the source code.

To retrieve source code for Amazon Linux packages

Amazon Web Services provides the source code for maintained packages. You can download the source
code for any installed packages with the **get_reference_source** command.

- Run the **get_reference_source -p** *package* command to download the source code for *package*.
  For example, to download the source code for the `htop` package, enter the following command.

  ```
  [ec2-user ~]$ get_reference_source -p htop

  Requested package: htop
  Found package from local RPM database: htop-1.0.1-2.3.amzn1.x86_64
  Corresponding source RPM to found package : htop-1.0.1-2.3.amzn1.src.rpm

  Are these parameters correct? Please type 'yes' to continue: yes
  Source RPM downloaded to: /usr/src/srpm/debug/htop-1.0.1-2.3.amzn1.src.rpm
  ```

The command output lists the location of the source RPM, in this case `/usr/src/srpm/de-bug/htop-1.0.1-2.3.amzn1.src.rpm`.

Managing User Accounts on Your Linux Instance

Each Linux instance type launches with a default Linux system user account. For Amazon Linux, the user name is `ec2-user`. For RHEL5, the user name is either `root` or `ec2-user`. For Ubuntu, the user name is `ubuntu`. For Fedora, the user name is either `fedora` or `ec2-user`. For SUSE Linux, the user name is `root`. Otherwise, if `ec2-user` and `root` don't work, check with your AMI provider.

Note

Linux system users should not be confused with AWS Identity and Access Management (IAM) users. For more information, see IAM Users and Groups in the *Using IAM* guide.

Using the default user account is adequate for many applications, but you may choose to add user accounts so that individuals can have their own files and workspaces. Creating user accounts for new users is much more secure than granting multiple (possibly inexperienced) users access to the `ec2-user` account, since that account can cause a lot of damage to a system when used improperly.

To add a new user to the system

Effectively adding users to a Linux instance involves two basic operations: adding the user to the system, and providing that user with a way to log in remotely.

1. To add a new user to the system, use the **adduser** command followed by any relevant options and the name of the user you wish to create.

    Important

    If you are adding a user to an Ubuntu system, you should add the `--disabled-password` option to avoid adding a password to the account.

    ```
    [ec2-user ~]$ sudo adduser newuser
    ```

    This command adds the `newuser` account to the system (with an entry in the `/etc/passwd` file), creates a `newuser` group, and creates a home directory for the account in `/home/newuser`.

2. To provide remote access to this account, you must create a `.ssh` directory in the `newuser` home directory and create a file within it named "`authorized_keys`" that contains a public key.

    a. Switch to the new account so that newly created files have the proper ownership.

        ```
        [ec2-user ~]$ sudo su - newuser
        [newuser ~]$
        ```

        Note that the prompt now says `newuser` instead of `ec2-user`; you have switched the shell session to the new account.

    b. Create a `.ssh` directory for the `authorized_keys` file.

        ```
        [newuser ~]$ mkdir .ssh
        ```

    c. Change the file permissions of the `.ssh` directory to `700` (this means only the file owner can read, write, or open the directory).

Important

This step is very important; without these exact file permissions, you will not be able to log into this account using SSH.

```
[newuser ~]$ chmod 700 .ssh
```

d. Create a file named "`authorized_keys`" in the `.ssh` directory.

```
[newuser ~]$ touch .ssh/authorized_keys
```

e. Change the file permissions of the `authorized_keys` file to `600` (this means only the file owner can read or write to the file).

Important

This step is very important; without these exact file permissions, you will not be able to log into this account using SSH.

```
[newuser ~]$ chmod 600 .ssh/authorized_keys
```

f. Edit the `authorized_keys` file with your favorite text editor and paste the public key for your key pair into the file, for example:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhD
FcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJh
JOI0iBXr
lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8Se
JtjnV3iAoG/cQk+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXP
kX4rWi+z7wB3Rb
BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

Note

For more information about creating a key pair or retrieving a public key from an existing key pair, see Amazon EC2 Key Pairs (p. 413)

You should now be able to log into the `newuser` account on your instance via SSH using the private key that matches the public key from Step 2.f (p. 331).

To remove a user from the system

If a user account is no longer needed, you can remove that account so that it may no longer be used.

- To delete a user account, the user's home directory, and the user's mail spool, execute the **userdel -r** command followed by the user name you wish to delete.

```
[ec2-user ~]$ sudo userdel -r olduser
```

Note

To keep the user's home directory and mail spool, omit the `-r` option.

Setting the Time for Your Linux Instance

A consistent and accurate time reference is crucial for many server tasks and processes. Most system logs include a time stamp that you can use to determine when problems occur and in what order the events take place. Network Time Protocol (NTP) is configured by default on Amazon Linux instances, and the system time is synchronized with a load-balanced pool of public servers on the Internet and set to the UTC time zone. For more information about NTP, go to http://www.ntp.org/.

Tasks

- Changing the Time Zone (p. 332)
- Configuring Network Time Protocol (NTP) (p. 333)

> **Important**
> These procedures are intended for use with Amazon Linux, but the commands and file locations are similar for Red Hat and CentOS. For more information about other distributions, see their specific documentation.

Changing the Time Zone

Amazon Linux instances are set to the UTC (Coordinated Universal Time) time zone by default, but you may wish to change the time on an instance to the local time or to another time zone in your network.

To change the time zone on an instance

1. Identify the time zone to use on the instance. The `/usr/share/zoneinfo` directory contains a hierarchy of time zone data files. Browse the directory structure at that location to find a file for your time zone.

    ```
    [ec2-user ~]$ ls /usr/share/zoneinfo
    Africa      Chile    GB      Indian       Mideast   posixrules  US
    America     CST6CDT  GB-Eire Iran         MST       PRC         UTC
    Antarctica  Cuba     GMT     iso3166.tab  MST7MDT   PST8PDT     WET
    Arctic      EET      GMT0    Israel       Navajo    right       W-SU
    ...
    ```

    Some of the entries at this location are directories (such as `America`), and these directories contain time zone files for specific cities. Find your city (or a city in your time zone) to use for the instance. In this example, you can use the time zone file for Los Angeles, `/usr/share/zoneinfo/America/Los_Angeles`.

2. Update the `/etc/sysconfig/clock` file with the new time zone.

    a. Open the `/etc/sysconfig/clock` file with your favorite text editor (such as **vim** or **nano**). You need to use **sudo** with your editor command because `/etc/sysconfig/clock` is owned by `root`.

    b. Locate the `ZONE` entry, and change it to the time zone file (omitting the `/usr/share/zoneinfo` section of the path). For example, to change to the Los Angeles time zone, change the `ZONE` entry to the following.

    ```
    ZONE="America/Los_Angeles"
    ```

    c. Save the file and exit the text editor.

3. Create a symbolic link between `/etc/localtime` and your time zone file so that the instance finds the time zone file when it references local time information.

```
[ec2-user ~]$ sudo ln -sf /usr/share/zoneinfo/America/Los_Angeles /etc/loc
altime
```

4. Reboot the system to pick up the new time zone information in all services and applications.

```
[ec2-user ~]$ sudo reboot
```

Configuring Network Time Protocol (NTP)

Network Time Protocol (NTP) is configured by default on Amazon Linux instances; however, an instance needs access to the Internet for the standard NTP configuration to work. The procedures in this section show how to verify that the default NTP configuration is working correctly. If your instance does not have access to the Internet, you need to configure NTP to query a different server in your private network to keep accurate time.

To verify that NTP is working properly

1. Use the **ntpstat** command to view the status of the NTP service on the instance.

```
[ec2-user ~]$ ntpstat
```

If your output resembles the output below, then NTP is working properly on the instance.

```
synchronised to NTP server (12.34.56.78) at stratum 3
   time correct to within 399 ms
   polling server every 64 s
```

If your output states, "`unsynchronised`", wait a minute and try again. The first synchronization may take a minute to complete.

If your output states, "`Unable to talk to NTP daemon. Is it running?`", you probably need to start the NTP service and enable it to automatically start at boot time.

2. (Optional) You can use the **ntpq -p** command to see a list of peers known to the NTP server and a summary of their state.

```
[ec2-user ~]$ ntpq -p
     remote           refid      st t when poll reach   delay   offset
jitter
==============================================================================
+lttleman.deekay 204.9.54.119     2 u   15  128  377    88.649    5.946
6.876
-bittorrent.tomh 91.189.94.4      3 u  133  128  377   182.673    8.001
1.278
*ntp3.junkemailf 216.218.254.202  2 u   68  128  377    29.377    4.726
11.887
+tesla.selinc.co 149.20.64.28     2 u   31  128  377    28.586   -1.215
1.435
```

To start and enable NTP

1. Start the NTP service with the following command.

```
[ec2-user ~]$ sudo service ntpd start
Starting ntpd:                                              [  OK  ]
```

2. Enable NTP to start at boot time with the **chkconfig** command.

```
[ec2-user ~]$ sudo chkconfig ntpd on
```

3. Verify that NTP is enabled with the following command.

```
[ec2-user ~]$ sudo chkconfig --list ntpd
ntpd            0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

Here **ntpd** is on in runlevels 2, 3, 4, and 5, which is correct.

To change NTP servers

You may decide not to use the standard NTP servers or you may need to use your own NTP server within your private network for instances that do not have Internet access.

1. Open the `/etc/ntp.conf` file in your favorite text editor (such as **vim** or **nano**). You need to use **sudo** with the editor command because `/etc/ntp.conf` is owned by `root`.

2. Find the `server` section, which defines the servers to poll for NTP configuration.

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 0.amazon.pool.ntp.org iburst
server 1.amazon.pool.ntp.org iburst
server 2.amazon.pool.ntp.org iburst
server 3.amazon.pool.ntp.org iburst
```

3. Comment out the servers you don't want to use by adding a "#" character to the beginning of those server definitions.

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#server 0.amazon.pool.ntp.org iburst
#server 1.amazon.pool.ntp.org iburst
#server 2.amazon.pool.ntp.org iburst
#server 3.amazon.pool.ntp.org iburst
```

4. Add an entry for each server to poll for time synchronization. You can use a DNS name for this entry or a dotted quad IP address (such as `10.0.0.254`).

```
server my-ntp-server.my-domain.com iburst
```

5. Restart the NTP service to pick up the new servers.

```
[ec2-user ~]$ sudo service ntpd start
Starting ntpd:                                            [  OK  ]
```

6. Verify that your new settings work and that NTP is functioning.

```
[ec2-user ~]$ ntpstat
synchronised to NTP server (64.246.132.14) at stratum 2
   time correct to within 99 ms
```

Changing the Hostname of Your Linux Instance

When you launch an instance, it is assigned a hostname that is a form of the private, internal IP address. A typical Amazon EC2 private DNS name looks something like this: `ip-12-34-56-78.us-west-2.compute.internal`, where the name consists of the internal domain, the service (in this case, `compute`), the region, and a form of the private IP address. Part of this hostname is displayed at the shell prompt when you log into your instance (for example, `ip-12-34-56-78`). Each time you stop and restart your Amazon EC2 instance (unless you are using an Elastic IP address), the public and private IP addresses change, and so does your public DNS name, system hostname, and shell prompt.

Important
These procedures are intended for use with Amazon Linux, but the commands and file locations are similar for Red Hat and CentOS. For more information about other distributions, see their specific documentation.

Changing the System Hostname

If you have a public DNS name registered for the IP address of your instance (such as `webserver.mydomain.com`), you can set the system hostname so your instance identifies itself as a part of that domain. This also changes the shell prompt so that it displays the first portion of this name instead of the hostname supplied by AWS (for example, `ip-12-34-56-78`). If you do not have a public DNS name registered, you can still change the hostname, but the process is a little different.

To change the system hostname to a public DNS name

Follow this procedure if you already have a public DNS name registered.

1. Open the `/etc/sysconfig/network` configuration file in your favorite text editor and change the `HOSTNAME` entry to reflect the fully qualified domain name (such as `webserver.mydomain.com`).

```
HOSTNAME=webserver.mydomain.com
```

2. Reboot the instance to pick up the new hostname.

```
[ec2-user ~]$ sudo reboot
```

3. Log into your instance and verify that the hostname has been updated. Your prompt should show the new hostname (up to the first ".") and the **hostname** command should show the fully qualified domain name.

```
[ec2-user@webserver ~]$ hostname
webserver.mydomain.com
```

To change the system hostname without a public DNS name

1. Open the `/etc/sysconfig/network` configuration file in your favorite text editor and change the
    `HOSTNAME` entry to reflect the desired system hostname (such as `webserver`).

    ```
    HOSTNAME=webserver.localdomain
    ```

2. Open the `/etc/hosts` file in your favorite text editor and change the entry beginning with `127.0.0.1`
    to match the example below, substituting your own hostname.

    ```
    127.0.0.1 webserver.localdomain webserver localhost localhost.localdomain
    ```

3. Reboot the instance to pick up the new hostname.

    ```
    [ec2-user ~]$ sudo reboot
    ```

4. Log into your instance and verify that the hostname has been updated. Your prompt should show
    the new hostname (up to the first ".") and the **hostname** command should show the fully qualified
    domain name.

    ```
    [ec2-user@webserver ~]$ hostname
    webserver.localdomain
    ```

Changing the Shell Prompt Without Affecting the Hostname

If you do not want to modify the hostname for your instance, but you would like to have a more useful
system name (such as `webserver`) displayed than the private name supplied by AWS (for example, `ip-
12-34-56-78`), you can edit the shell prompt configuration files to display your system nickname instead
of the hostname.

To change the shell prompt to a host nickname

1. Create a file in `/etc/profile.d` that sets the environment variable called `NICKNAME` to the value
    you want in the shell prompt. For example, to set the system nickname to `webserver`, execute the
    following command.

    ```
    [ec2-user ~]$ sudo sh -c 'echo "export NICKNAME=webserver" > /etc/pro
    file.d/prompt.sh'
    ```

2. Open the `/etc/bashrc` file in your favorite text editor (such as **vim** or **nano**). You need to use **sudo**
    with the editor command because `/etc/bashrc` is owned by `root`.

3.   Edit the file and change the shell prompt variable (`PS1`) to display your nickname instead of the
     hostname. Find the following line that sets the shell prompt in `/etc/bashrc` (several surrounding
     lines are shown below for context; look for the line that starts with `[  "$PS1")`:

```
# Turn on checkwinsize
shopt -s checkwinsize
[ "$PS1" = "\\s-\\v\\\$ " ] && PS1="[\u@\h \W]\\$ "
# You might want to have e.g. tty in prompt (e.g. more virtual machines)
# and console windows
```

And change the `\h` (the symbol for `hostname`) in that line to the value of the `NICKNAME` variable.

```
# Turn on checkwinsize
shopt -s checkwinsize
[ "$PS1" = "\\s-\\v\\\$ " ] && PS1="[\u@$NICKNAME \W]\\$ "
# You might want to have e.g. tty in prompt (e.g. more virtual machines)
# and console windows
```

4.   (Optional) To set the title on shell windows to the new nickname, complete the following steps.

     a.   Create a file called `/etc/sysconfig/bash-prompt-xterm`.

     ```
     [ec2-user ~]$ sudo touch /etc/sysconfig/bash-prompt-xterm
     ```

     b.   Make the file executable with the following command.

     ```
     [ec2-user ~]$ sudo chmod +x /etc/sysconfig/bash-prompt-xterm
     ```

     c.   Open the `/etc/sysconfig/bash-prompt-xterm` file in your favorite text editor (such as **vim**
          or **nano**). You need to use **sudo** with the editor command because `/etc/sysconfig/bash-
          prompt-xterm` is owned by `root`.
     d.   Add the following line to the file.

     ```
     echo -ne "\033]0;${USER}@${NICKNAME}:${PWD/#$HOME/~}\007"
     ```

5.   Log out and then log back in to pick up the new nickname value.

Setting Up Dynamic DNS on Your Linux Instance

When you launch an Amazon EC2 instance, it is assigned a public IP address and a public DNS (Domain
Name System) name that you can use to reach it from the Internet. Because there are so many hosts in
the Amazon Web Services domain, these public names must be quite long for each name to remain
unique. A typical Amazon EC2 public DNS name looks something like this: `ec2-12-34-56-78.us-
west-2.compute.amazonaws.com`, where the name consists of the Amazon Web Services domain,
the service (in this case, `compute`), the region, and a form of the public IP address.

Dynamic DNS services provide custom DNS host names within their domain area that can be easy to
remember and that can also be more relevant to your host's use case; some of these services are also
free of charge. You can use a dynamic DNS provider with Amazon EC2 and configure the instance to

update the IP address associated with a public DNS name each time the instance starts. There are many different providers to choose from, and the specific details of choosing a provider and registering a name with them are outside the scope of this guide. A simple web search of "dynamic DNS" should return multiple results for service providers, many of whom provide this service for free.

Important

These procedures are intended for use with Amazon Linux, but the commands and file locations are similar for Red Hat and CentOS. For more information about other distributions, see their specific documentation.

To use dynamic DNS with Amazon EC2

1. Sign up with a dynamic DNS service provider and register a public DNS name with their service. In this procedure, you can use the free service from no-ip.com/free.

2. Configure the dynamic DNS update client. After you have a dynamic DNS service provider and a public DNS name registered with their service, point the DNS name to the IP address for your instance. Many providers allow you to do this manually from your account page on their website, but many also allow you to do this automatically with a software update client. In this example, you can use the `noip2` client, which only works with the service provided by no-ip.com.

   Note

   Other service providers may offer their own client, or you may be able to configure the `ddclient` update utility to work with their service. For more information, see the specific documentation for your service provider and http://sourceforge.net/p/ddclient/wiki/Home/.

   a. Enable the Extra Packages for Enterprise Linux (EPEL) repository to gain access to the `noip` client. The `ddclient` package is also available in this repository, but the configuration steps that follow are different.

      Note

      Amazon Linux instances have the GPG keys and repository information for the EPEL repository installed by default; however, Red Hat and CentOS instances must first install the `epel-release` package before you can enable the EPEL repository. For more information and to download the latest version of this package, see https://fedoraproject.org/wiki/EPEL.

      ```
      [ec2-user ~]$ sudo yum-config-manager --enable epel
      ```

   b. Install the `noip` package.

      ```
      [ec2-user ~]$ sudo yum install -y noip
      ```

   c. Create the `noip2` configuration file. Enter the login and password information for when prompted and answer the subsequent questions to configure the client.

      ```
      [ec2-user ~]$ sudo noip2 -C
      ```

3. Enable the `noip` service with the **chkconfig** command.

   ```
   [ec2-user ~]$ sudo chkconfig noip on
   ```

   You can verify that the service is enabled with the **chkconfig --list** command.

```
[ec2-user ~]$ chkconfig --list noip
noip            0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

Here, **noip** is `on` in runlevels 2, 3, 4, and 5 (which is correct). Now the update client starts at every boot and updates the public DNS record to point to the IP address of the instance.

4. Start the `noip` service.

```
[ec2-user ~]$ sudo service noip start
Starting noip2:                                          [  OK  ]
```

This command starts the client, which reads the configuration file (`/etc/no-ip2.conf`) that you created earlier and updates the IP address for the public DNS name that you chose.

5. Verify that the update client has set the correct IP address for your dynamic DNS name. Allow a few minutes for the DNS records to update, and then try to connect to your instance via SSH with the public DNS name that you configured in this procedure.

Running Commands on Your Linux Instance at Launch

When you launch an instance in Amazon EC2, you have the option of passing user data to the instance that can be used to perform common automated configuration tasks and even run scripts after the instance starts. You can pass two types of user data to Amazon EC2: shell scripts and `cloud-init` directives. You can also pass this data into the launch wizard as plain text, as a file (this is useful for launching instances via the command line tools), or as base64-encoded text (for API calls).

If you are interested in more complex automation scenarios, consider using AWS CloudFormation and AWS OpsWorks. For more information, see the AWS CloudFormation User Guide and the AWS OpsWorks User Guide.

In the following examples, the commands from the Installing a LAMP Web Server tutorial (p. 38) are converted to a shell script and a set of `cloud-init` directives that executes when the instance launches. In each example, the following tasks are executed by the user data:

*   The distribution software packages are updated.
*   The necessary web server, `php`, and `mysql` packages are installed.
*   The `httpd` service is started and turned on via **chkconfig**.
*   The `www` group is added, and the `ec2-user` is added to that group.
*   The appropriate ownership and file permissions are set for the web directory and the files contained within it.
*   A simple web page is created to test the web server and `php` engine.

Contents

Prerequisites

The following examples assume that your instance has a public DNS name that is reachable from the Internet. For more information, see Launch an Amazon EC2 Instance (p. 27). You must also configure your security group to allow SSH (port 22), HTTP (port 80), and HTTPS (port 443) connections. For more information about these prerequisites, see Setting Up with Amazon EC2 (p. 20).

Also, these instructions are intended for use with Amazon Linux, and the commands and directives may not work for other Linux distributions. For more information about other distributions, such as their support for cloud-init, see their specific documentation.

User Data and Shell Scripts

If you are familiar with shell scripting, this is the easiest and most complete way to send instructions to an instance at launch, and the cloud-init log file (/var/log/cloud-init.log) captures console output so it is easy to debug your scripts following a launch if the instance does not behave the way you intended.

> **Important**
> User data scripts and cloud-init directives only run during the first boot cycle when an instance is launched.

User data shell scripts must start with the #! characters and the path to the interpreter you want to read the script (commonly **/bin/bash**). For a great introduction on shell scripting, see the BASH Programming HOW-TO at the Linux Documentation Project (tldp.org).

Scripts entered as user data are executed as the root user, so do not use the **sudo** command in the script. Remember that any files you create will be owned by root; if you need non-root users to have file access, you should modify the permissions accordingly in the script. Also, because the script is not run interactively, you cannot include commands that require user feedback (such as **yum update** without the -y flag).

Adding these tasks at boot time adds to the amount of time it takes to boot the instance. You should allow a few minutes of extra time for the tasks to complete before you test that the user script has finished successfully.

To pass a shell script to an instance with user data

1. Follow the procedure for launching an instance at Launching Your Instance from an AMI (p. 290), but when you get to Step 6 (p. 292), paste the user data script text into the **User data** field and then complete the launch procedure. For the example below, the script creates and configures our web server.

    ```
    #!/bin/bash
    yum update -y
    yum groupinstall -y "Web Server" "MySQL Database" "PHP Support"
    yum install -y php-mysql
    service httpd start
    chkconfig httpd on
    groupadd www
    usermod -a -G www ec2-user
    chown -R root:www /var/www
    chmod 2775 /var/www
    find /var/www -type d -exec chmod 2775 {} +
    find /var/www -type f -exec chmod 0664 {} +
    echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
    ```

2. Allow enough time for the instance to launch and execute the commands in your script, and then check to see that your script has completed the tasks that you intended. For our example, in a web browser, enter the URL of the PHP test file the script created. This URL is the public DNS address of your instance followed by a forward slash and the file name.

    ```
    http://my.public.dns.amazonaws.com/phpinfo.php
    ```

    You should see the PHP information page.

    > **Tip**
    > If you are unable to see the PHP information page, check that the security group you are using contains a rule to allow HTTP (port 80) traffic. For information about adding an HTTP rule to your security group, see Adding Rules to a Security Group (p. 426).

3. (Optional) If your script did not accomplish the tasks you were expecting it to, or if you just want to verify that your script completed without errors, examine the cloud-init log file at /var/log/cloud-init.log and look for error messages in the output.

    For additional debugging information, you can create a Mime multipart archive that includes a cloud-init data section with the following directive:

    ```
    output : { all : '| tee -a /var/log/cloud-init-output.log' }
    ```

    This directive sends command output from your script to /var/log/cloud-init-output.log. For more information on cloud-init data formats and creating Mime multi part archive, see cloud-init Formats.

User Data and cloud-init Directives

The cloud-init package configures specific aspects of a new Amazon Linux instance when it is launched; most notably, it configures the .ssh/authorized_keys file for the ec2-user so you can log in with your own private key.

The cloud-init user directives can be passed to an instance at launch the same way that a script is passed, although the syntax is different. For more information about cloud-init, go to http://cloudin-it.readthedocs.org/en/latest/index.html.

> **Important**
> User data scripts and cloud-init directives only run during the first boot cycle when an instance is launched.

The Amazon Linux version of cloud-init does not support all of the directives that are available in the base package, and some of the directives have been renamed (such as repo_update instead of apt-upgrade).

Adding these tasks at boot time adds to the amount of time it takes to boot an instance. You should allow a few minutes of extra time for the tasks to complete before you test that your user data directives have completed.

To pass cloud-init directives to an instance with user data

1. Follow the procedure for launching an instance at Launching Your Instance from an AMI (p. 290), but when you get to Step 6 (p. 292), paste your cloud-init directive text into the **User data** field and then complete the launch procedure. For the example below, the directives create and configure a web server.

```
#cloud-config
repo_update: true
repo_upgrade: all

packages:
 - httpd
 - crypto-utils
 - mod_perl
 - mod_ssl
 - mod_wsgi
 - mysql-server
 - php
 - php-gd
 - php-pdo
 - php-pear
 - php-xml
 - php-mysql

runcmd:
 - service httpd start
 - chkconfig httpd on
 - groupadd www
 - [ sh, -c, "usermod -a -G www ec2-user" ]
 - [ sh, -c, "chown -R root:www /var/www" ]
 - chmod 2775 /var/www
 - [ find, /var/www, -type, d, -exec, chmod, 2775, {}, + ]
 - [ find, /var/www, -type, f, -exec, chmod, 0664, {}, + ]
 - [ sh, -c, 'echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php' ]
```

2. Allow enough time for the instance to launch and execute the directives in your user data, and then check to see that your directives have completed the tasks you intended. For our example, in a web browser, enter the URL of the PHP test file the directives created. This URL is the public DNS address of your instance followed by a forward slash and the file name.

```
http://my.public.dns.amazonaws.com/phpinfo.php
```

You should see the PHP information page.

> **Tip**
> If you are unable to see the PHP information page, check that the security group you are using contains a rule to allow HTTP (port 80) traffic. For information about adding an HTTP rule to your security group, see Adding Rules to a Security Group (p. 426).

3. (Optional) If your directives did not accomplish the tasks you were expecting them to, or if you just want to verify that your directives completed without errors, examine the cloud-init log file at /var/log/cloud-init.log and look for error messages in the output. For additional debugging information, you can add the following line to your directives:

```
output : { all : '| tee -a /var/log/cloud-init-output.log' }
```

This directive sends **runcmd** output to /var/log/cloud-init-output.log.

Monitoring Amazon EC2

Monitoring is an important part of maintaining the reliability, availability, and performance of your Amazon Elastic Compute Cloud (Amazon EC2) instances and your AWS solutions. You should collect monitoring data from all of the parts in your AWS solutions so that you can more easily debug a multi-point failure if one occurs. Before you start monitoring Amazon EC2, however, you should create a monitoring plan that should include:

- What are your goals for monitoring?
- What resources you will monitor?
- How often you will monitor these resources?
- What monitoring tools will you use?
- Who will perform the monitoring tasks?
- Who should be notified when something goes wrong?

After you have defined your monitoring goals and have created your monitoring plan, the next step is to establish a baseline for normal Amazon EC2 performance in your environment. You should measure Amazon EC2 performance at various times and under different load conditions. As you monitor Amazon EC2, you should store a history of monitoring data that you've collected. You can compare current Amazon EC2 performance to this historical data to help you to identify normal performance patterns and performance anomalies, and devise methods to address them. For example, you can monitor CPU utilization, disk I/O, and network utilization for your Amazon EC2 instances. When performance falls outside your established baseline, you might need to reconfigure or optimize the instance to reduce CPU utilization, improve disk I/O, or reduce network traffic.

To establish a baseline you should, at a minimum, monitor the following items:

Item to Monitor	Amazon EC2 Metric	Monitoring Script
CPU utilization	CPUUtilization (p. 358)	
Memory utilization		Monitoring Scripts for Amazon EC2 Instances (p. 405)
Memory used		Monitoring Scripts for Amazon EC2 Instances (p. 405)
Memory available		Monitoring Scripts for Amazon EC2 Instances (p. 405)

Item to Monitor	Amazon EC2 Metric	Monitoring Script
Network utilization	NetworkIn (p. 358) NetworkOut (p. 358)	
Disk performance	DiskReadOps (p. 358) DiskWriteOps (p. 358)	
Disk Swap utilization Swap used (Linux)		Monitoring Scripts for Amazon EC2 Instances (p. 405)
Page File utilization Page File used Page File available		Monitoring Scripts for Amazon EC2 Instances (p. 405)
Disk Reads/Writes	DiskReadBytes (p. 358) DiskWriteBytes (p. 358)	
Disk Space utilization		Monitoring Scripts for Amazon EC2 Instances (p. 405)
Disk Space used		Monitoring Scripts for Amazon EC2 Instances (p. 405)
Disk Space available		Monitoring Scripts for Amazon EC2 Instances (p. 405)

Automated and Manual Monitoring

AWS provides various tools that you can use to monitor Amazon EC2. You can configure some of these tools to do the monitoring for you, while some of the tools require manual intervention.

Topics

Automated Monitoring Tools

You can use the following automated monitoring tools to watch Amazon EC2 and report back to you when something is wrong:

- **System Status Checks** - monitor the AWS systems required to use your instance to ensure they are working properly. These checks detect problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue or you can resolve it yourself (for example, by stopping and restarting or terminating and replacing an instance). Examples of problems that cause system status checks to fail include:
  - Loss of network connectivity
  - Loss of system power
  - Software issues on the physical host

- Hardware issues on the physical host

For more information, see Monitoring Instances with Status Checks (p. 346).

- **Instance Status Checks** - monitor the software and network configuration of your individual instance. These checks detect problems that require your involvement to repair. When an instance status check fails, typically you will need to address the problem yourself (for example by rebooting the instance or by making modifications in your operating system). Examples of problems that may cause instance status checks to fail include:
  - Failed system status checks
  - Misconfigured networking or startup configuration
  - Exhausted memory
  - Corrupted file system
  - Incompatible kernel

  For more information, see Monitoring Instances with Status Checks (p. 346).

- **Amazon CloudWatch Alarms** - watch a single metric over a time period you specify, and perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon Simple Notification Service (Amazon SNS) topic or Auto Scaling policy. Alarms invoke actions for sustained state changes only. CloudWatch alarms will not invoke actions simply because they are in a particular state, the state must have changed and been maintained for a specified number of periods. For more information, see Monitoring Your Instances with CloudWatch (p. 354).

- **Amazon EC2 Monitoring Scripts** - Perl and PowerShell scripts that can monitor memory, disk, and page/swap file usage in your instances. For more information, see Monitoring Scripts for Amazon EC2 Instances (p. 405).

- **AWS Management Pack for Microsoft System Center Operations Manager** - links Amazon EC2 instances and the Microsoft Windows or Linux operating systems running inside them. The AWS Management Pack is an extension to Microsoft System Center Operations Manager. It uses a designated computer in your datacenter (called a watcher node) and the Amazon Web Services APIs to remotely discover and collect information about your AWS resources. For more information, see AWS Management Pack for Microsoft System Center.

Manual Monitoring Tools

Another important part of monitoring Amazon EC2 involves manually monitoring those items that the monitoring scripts, status checks, and CloudWatch alarms don't cover. The Amazon EC2 and CloudWatch console dashboards provide an at-a-glance view of the state of your Amazon EC2 environment.

- Amazon EC2 Dashboard shows:
  - Service Health and Scheduled Events by region
  - Instance state
  - Status checks
  - Alarm status
  - Instance metric details (In the navigation pane click **Instances**, select an instance, and then click the **Monitoring** tab)
  - Volume metric details (In the navigation pane click **Volumes**, select a volume, and then click the **Monitoring** tab)
- Amazon CloudWatch Dashboard shows:
  - Current alarms and status
  - Graphs of alarms and resources
  - Service health status

In addition, you can use CloudWatch to do the following:

- Graph Amazon EC2 monitoring data to troubleshoot issues and discover trends
- Search and browse all your AWS resource metrics
- Create and edit alarms to be notified of problems
- See at-a-glance overviews of your alarms and AWS resources

Best Practices for Monitoring

Use the following best practices for monitoring to help you with your Amazon EC2 monitoring tasks.

- Make monitoring a priority to head off small problems before they become big ones.
- Create and implement a monitoring plan that collects monitoring data from all of the parts in your AWS solution so that you can more easily debug a multi-point failure if one occurs. Your monitoring plan should address, at a minimum, the following questions:
  - What are your goals for monitoring?
  - What resources you will monitor?
  - How often you will monitor these resources?
  - What monitoring tools will you use?
  - Who will perform the monitoring tasks?
  - Who should be notified when something goes wrong?
- Automate monitoring tasks as much as possible.
- Check the log files on your EC2 instances.

Monitoring the Status of Your Instances

You can monitor the status of your instances by viewing status checks and scheduled events for your instances. A status check gives you the information that results from automated checks performed by Amazon EC2. These automated checks detect whether specific issues are affecting your instances. The status check information, together with the data provided by Amazon CloudWatch, gives you detailed operational visibility into each of your instances.

You can also see status on specific events scheduled for your instances. Events provide information about upcoming activities such as rebooting or retirement that are planned for your instances, along with the scheduled start and end time of each event.

Contents

Monitoring Instances with Status Checks

With instance status monitoring you can quickly determine whether Amazon EC2 has detected any problems that may prevent your instances from running applications. Amazon EC2 performs automated checks on every running Amazon EC2 instance to identify hardware and software issues. You can view the results of these status checks to identify specific and detectable problems. This data augments the information that Amazon EC2 already provides about the intended state of each instance (pending, running, stopping, etc.) as well as the utilization metrics that Amazon CloudWatch monitors (CPU utilization, network traffic, and disk activity).

Status checks are performed every minute and each returns a pass or a fail status. If all checks pass, the overall status of the instance is **OK**. If one or more checks fail, the overall status is **impaired**. Status checks are built into Amazon EC2, so they cannot be disabled or deleted. You can, however create or delete alarms that are triggered based on the result of the status checks. For example, you can create an alarm to warn you if status checks fail on a specific instance. For more information, see Creating and Editing Status Check Alarms (p. 349).

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

Viewing Status

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

Amazon EC2 Console

To view status checks using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Instances**.
3. On the **Instances** page, the **Status Checks** column lists the operational status of each instance.
4. To view an individual instance's status, select the instance, and then click the **Status Checks** tab.

Note

If you have an instance with a failed status check and the instance has been unreachable for over 20 minutes, you can click **Contact AWS Support** to submit a request for assistance. To try and troubleshoot system or instance status check failures yourself, see Troubleshooting Instances with Failed Status Checks (p. 639).

Command Line Interface

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

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

Note

If you have an instance with a failed status check, see Troubleshooting Instances with Failed Status Checks (p. 639)

API

You can use the `DescribeInstanceStatus` action to retrieve the status of your instances. For more information, see DescribeInstanceStatus in the *Amazon EC2 API Reference*.

Reporting Status

You can provide feedback about your instances if you are having problems with an instance whose status is not shown as impaired, or to send AWS additional details about the problems you are experiencing with an impaired instance.

We use reported feedback to identify issues impacting multiple customers, but do not respond to individual account issues reported via this form. Providing feedback does not change the status check results that you currently see for this instance.

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

Amazon EC2 Console

To report status feedback using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Instances**.
3. On the **Instances** page, click on the instance on which you want to report status.
4. Click the **Status Checks** tab, and then click **Submit feedback**.
5. Complete the information on the **Report Instance Status** page.

Command Line Interface

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

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

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

API

You can use the `ReportInstanceStatus` action to submit feedback about a running instance's status. If your experience with the instance differs from the instance status returned by the `DescribeIn-stanceStatus` 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 EC2 API Reference*.

Creating and Editing Status Check Alarms

You can create instance status and system status alarms to notify you when an instance has a failed status check. To create or change these alarms, you can use either the AWS Management Console or the command line interface (CLI).

AWS Management Console

To create a status check alarm

You can create status check alarms for an existing instance to monitor instance status or system status. You can configure the alarm to send you a notification by email when an instance fails an instance check or system status check.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Instances**.
3. Select an instance, and then on the **Status Checks** tab, click **Create Status Check Alarm**.
4. In the **Create Alarm** dialog box, select the **Send a notification to** check box, and then choose an existing Amazon Simple Notification Service (SNS) topic or create a new SNS topic to use for this alarm.
5. In the **With these recipients** box, type your email address (e.g., john.stiles@example.com) and the addresses of any additional recipients, separated by commas.
6. In the **Whenever** drop-down list, select the status check you want to be notified about (e.g., Status Check Failed (Any), Status Check Failed (Instance), or Status Check Failed (System)).

7.   In the **For at least** box, set the number of periods you want to evaluate (for example, 2) and in the **consecutive periods** drop-down menu, select the evaluation period duration (for example, 5 minutes) before triggering the alarm and sending an email

8.   To change the default name for the alarm, in the **Name of alarm** box, type a friendly name for the alarm (for example, StatusCheckFailed), and then click **Create Alarm**.

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

To edit a status check alarm

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

1.   Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2.   In the navigation pane, click **Instances**.
3.   Select an instance, click **Actions**, and then click **Add/Edit Alarms**.
4.   In the **Alarm Details** dialog box, click the name of the alarm.
5.   In the **Edit Alarm** dialog box, make the desired changes, and then click **Save**.

Command Line Interface

To create a status check alarm using the CLI

You can create a status check alarm using the AWS CLI. In the following example, the alarm publishes a notification to a specific SNS topic that has the ARN arn:aws:sns:us-east-1:1111111111:StatusCheckNotifications when instance i-ab12345 fails either the instance check or system status check for at least two periods. (The metric is StatusCheckFailed.) For more information, see the put-metric-alarm command in the *AWS Command Line Interface Reference*.

1.   At a command prompt, type `aws cloudwatch list-metrics` to view the list of all available Amazon CloudWatch metrics for the services in AWS that you're using.

2.   In the list of metrics, review the Status Check metrics that have the **AWS/EC2** namespace. These are the status check metrics that you can use to create a status check alarm.

3.   At the command prompt, enter the following command:

```
$ aws cloudwatch put-metric-alarm
                         --alarm-name StatusCheckFailed-Alarm-for-i-
ab12345 --alarm-description "Alarm when StatusCheckFailed metric has a value
 of one for two periods" --metric-name StatusCheckFailed --namespace AWS/EC2
 --statistic Maximum --dimensions Name=InstanceId,Value=i-ab12345 --period
 300 --unit Count --evaluation-periods 2 --threshold 1 --comparison-operator
 GreaterThanThreshold --alarm-actions arn:aws:sns:us-east-
1:1111111111StatusCheckNotifications
```

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, StatusCheck-Failed_Instance, or StatusCheckFailed_System). This is required.

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

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

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

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

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

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

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

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

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

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

Monitoring Events for Your Instances

Instance status describes specific events that AWS may schedule for your instances, such as a reboot or retirement. These scheduled events are not frequent. If one of your instances will be affected by a scheduled event, the email address that's associated with your AWS account receives an email prior to the scheduled event with details about the event, as well as a start and end date. (To update the contact information for your account, go to the Account Settings page.) You can also view scheduled events for your instance by using the Amazon EC2 console, API, or CLI. For more information, see Viewing Scheduled Events (p. 352).

There are different types of scheduled events:

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

If one of your instances is scheduled for any of the above events, you may be able to take actions to control the timing of the event, or to minimize downtime. For more information, see Working with an Instance That Has a Scheduled Event (p. 352).

Contents
- Viewing Scheduled Events (p. 352)

Viewing Scheduled Events

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

Amazon EC2 Console

To view scheduled events for your instances using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Events**. You can see a list of all resources with events associated with them. You can filter by instance or volume, or by specific status types.
3. Alternatively, you can do the following to view upcoming scheduled events:

   a. In the navigation pane, click the **EC2 Dashboard**.

   b. Under **Scheduled Events**, you can see the events associated with your Amazon EC2 instances and volumes.

Command Line Interface and API

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

Working with an Instance That Has a Scheduled Event

If your instance has a scheduled event, your course of action will depend on whether your instance's root device volume is an Amazon EBS volume or an instance store volume. You can determine the root device type for an instance by checking the value of the **Root device type** field in the details pane on the **Instances** page.

Instances Scheduled for Reboot

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

To view the type of scheduled reboot events

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

Actions Required for System Reboot

No action is required on your part if one of your instances is scheduled for a system reboot - you can wait for the reboot to occur automatically within its scheduled maintenance window. However, you can also reboot your instance manually at a time that is convenient for you before the reboot event is scheduled to begin. For more information, see Reboot Your Instance (p. 312).

The reboot typically completes in a matter of minutes. After the reboot completes, you can begin using your instance again. 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 whether the software on your instance is operating as you expect.

Actions Required for Instance Reboot

No action is required on your part if one of your instances is scheduled for an instance reboot - you can wait for the reboot to occur automatically within its scheduled maintenance window. However, you can also reboot your instance manually at a time that is convenient for you before the reboot event is scheduled to begin. For more information, see Reboot Your Instance (p. 312).

After you reboot your instance, the scheduled event for the instance reboot is canceled immediately and the event's description is updated. The pending maintenance to the underlying host is completed, and you can begin using your instance again after it has fully booted.

Instances Scheduled to Be Stopped or Retired

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

Actions Required for Instances Scheduled to Be Stopped or Retired

If your instance's root device is an Amazon EBS volume, you can wait for the instance to be stopped or retired at the scheduled event start time. Alternatively, you can stop and start the instance yourself. Doing so migrates your instance to new hardware and help reduce unforeseen downtime. For more information about stopping your instance, as well as information about changes to your instance configuration when it's stopped, see Stop and Start Your Instance (p. 309).

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

For more information about instances scheduled for retirement and how to manage them, see Instance Retirement (p. 312).

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

Instances Scheduled for Maintenance

Instances are scheduled for maintenance when underlying Amazon EC2 hardware requires maintenance. There are two types of maintenance events: network maintenance and power maintenance. During network maintenance, scheduled instances lose network connectivity for a brief period of time. Normal network connectivity to your instance will be restored after maintenance is complete. During power maintenance, scheduled instances are offline for a brief period, and then rebooted.

Actions Required for Instances Scheduled for Maintenance

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

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

Monitoring Your Instances with CloudWatch

You can monitor your Amazon EC2 instances using Amazon CloudWatch, which collects and processes raw data from Amazon EC2 into readable, near real-time metrics. These statistics are recorded for a period of two weeks, so that you can access historical information and gain a better perspective on how your web application or service is performing. By default, Amazon EC2 metric data is automatically sent to CloudWatch in 5-minute periods. You can, however, enable detailed monitoring on an Amazon EC2 instance, which sends data to CloudWatch in 1-minute periods. For more information about Amazon CloudWatch, see the Amazon CloudWatch Developer Guide.

The following table describes basic and detailed monitoring for Amazon EC2 instances.

Type	Description
Basic	Data is available automatically in 5-minute periods at no charge.
Detailed	Data is available in 1-minute periods at an additional cost. To get this level of data, you must specifically enable it for the instance. For the instances where you've enabled detailed monitoring, you can also get aggregated data across groups of similar instances. For information about pricing, see the Amazon CloudWatch product page.

You can get monitoring data for your Amazon EC2 instances using either the Amazon CloudWatch API or the AWS Management Console. The console displays a series of graphs based on the raw data from the Amazon CloudWatch API. Depending on your needs, you might prefer to use either the data from the API or the graphs in the console.

Contents

Enabling or Disabling Detailed Monitoring on an Amazon EC2 Instance

This section describes how to enable or disable detailed monitoring on either a new instance (as you launch it) or on a running or stopped instance. After you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period for the instance. You can enable or disable detailed monitoring using the console or the command line interface (CLI).

AWS Management Console

To enable detailed monitoring of an existing EC2 instance

You can enable detailed monitoring of your EC2 instances, which provides data about your instance in 1-minute periods. (There is an additional charge for 1-minute monitoring.) Detailed data is then available for the instance in the AWS Management Console graphs or through the API. To get this level of data, you must specifically enable it for the instance. For the instances on which you've enabled detailed monitoring, you can also get aggregated data across groups of similar instances. An instance must be running or stopped to enable detailed monitoring.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Instances**.
3. In the list of instances, select a running or stopped instance, click **Actions**, and then click **Enable Detailed Monitoring**.
4. In the **Enable Detailed Monitoring** dialog box, click **Yes, Enable**.
5. In the **Enable Detailed Monitoring** confirmation dialog box, click **Close**.

   Detailed data (collected with a 1-minute period) is then available for the instance in the AWS Management Console graphs or through the API.

To enable detailed monitoring when launching an EC2 instance

When launching an instance with the AWS Management Console, select the **Monitoring** check box on the **Configure Instance Details** page of the launch wizard.

After the instance is launched, you can select the instance in the console and view its monitoring graphs on the instance's **Monitoring** tab in the lower pane.

To disable detailed monitoring of an EC2 instance

When you no longer want to monitor your instances at 1-minute intervals, you can disable detailed monitoring and use basic monitoring instead. Basic monitoring provides data in 5-minute periods at no charge.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Instances**.
3. In the list of instances, select a running or stopped instance, click **Actions**, and then click **Disable Detailed Monitoring**.
4. In the **Disable Detailed Monitoring** dialog box, click **Yes, Disable**.
5. In the **Disable Detailed Monitoring** confirmation dialog box, click **Close**.

For information about launching instances, see .

Command Line Interface

To enable detailed monitoring on an existing instance

Use the `monitor-instances` command with one or more instance IDs. For more information about using the **monitor-instances** command, see monitor-instances in the *AWS Command Line Interface Reference*.

```
$ aws ec2 monitor-instances --instance-ids i-570e5a28
{
    "InstanceMonitorings": [
        {
            "InstanceId": "i-570e5a28",
            "Monitoring": {
                "State": "pending"
            }
        }
    ]
}
```

Detailed data (collected with a 1-minute period) is then available for the instance in the AWS Management Console graphs or through the API.

To enable detailed monitoring when launching an instance

Use the `run-instances` command with the `--monitoring` flag. For more information about using the **run-instances** command, see run-instances in the *AWS Command Line Interface Reference*.

```
$ aws ec2 run-instances --image-id ami-09092360 --key-name MyKeyPair --monitoring
 Enabled=value
```

Amazon EC2 returns output similar to the following example. The status of monitoring is listed as *pending*.

```
{
    "OwnerId": "111122223333",
    "ReservationId": "r-25fad905",
    "Groups": [
        {
            "GroupName": "default",
            "GroupId": "sg-eafe1b82"
        }
    ],
    "Instances": [
        {
            "Monitoring": {
                "State": "pending"
            },
            "PublicDnsName": null,
            "Platform": "windows",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "EbsOptimized": false,
            "LaunchTime": "2014-02-24T18:02:49.000Z",
            "ProductCodes": [],
```

```
                "StateTransitionReason": null,
                "InstanceId": "i-31283b11",
                "ImageId": "ami-09092360",
                "PrivateDnsName": null,
                "KeyName": "MyKeyPair",
                "SecurityGroups": [
                    {
                        "GroupName": "default",
                        "GroupId": "sg-eafe1b82"
                    }
                ],
                "ClientToken": null,
                "InstanceType": "m1.small",
                "NetworkInterfaces": [],
                "Placement": {
                    "Tenancy": "default",
                    "GroupName": null,
                    "AvailabilityZone": "us-east-1b"
                },
                "Hypervisor": "xen",
                "BlockDeviceMappings": [],
                "Architecture": "x86_64",
                "StateReason": {
                    "Message": "pending",
                    "Code": "pending"
                },
                "VirtualizationType": "hvm",
                "RootDeviceType": "instance-store",
                "AmiLaunchIndex": 0
            }
        ]
}
```

After the instance is running, detailed data (collected with a 1-minute period) is then available for the instance in the AWS Management Console graphs or through the API.

To disable detailed monitoring of an instance

Use the `unmonitor-instances` command with one or more instance IDs. For more information about using the **unmonitor-instances** command, see unmonitor-instances in the *AWS Command Line Interface Reference*.

```
$ aws ec2 unmonitor-instances --instance-ids i-570e5a28
{
    "InstanceMonitorings": [
        {
            "InstanceId": "i-570e5a28",
            "Monitoring": {
                "State": "disabling"
            }
        }
    ]
}
```

View Amazon EC2 Metrics

Only those services in AWS that you're using send metrics to Amazon CloudWatch. You can use the Amazon CloudWatch console, the `mon-list-metrics` command, or the `ListMetrics` API to view the metrics that Amazon EC2 sends to CloudWatch. If you've enabled detailed monitoring, each data point covers the instance's previous 1 minute of activity. Otherwise, each data point covers the instance's previous 5 minutes of activity.

Metric	Description
`CPUCreditUsage`	(Only valid for T2 instances) The number of CPU credits consumed during the specified period. This metric identifies the amount of time during which physical CPUs were used for processing instructions by virtual CPUs allocated to the instance. **Note** CPU Credit metrics are available at a 5 minute frequency. Units: Count
`CPUCreditBalance`	(Only valid for T2 instances) The number of CPU credits that an instance has accumulated. This metric is used to determine how long an instance can burst beyond its baseline performance level at a given rate. **Note** CPU Credit metrics are available at a 5 minute frequency. Units: Count
`CPUUtilization`	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. Units: Percent
`DiskReadOps`	Completed read operations from all ephemeral disks available to the instance in a specified period of time. If your instance uses Amazon EBS volumes, see Amazon EBS Metrics (p. 535). **Note** To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period. Units: Count
`DiskWriteOps`	Completed write operations to all ephemeral disks available to the instance in a specified period of time. If your instance uses Amazon EBS volumes, see Amazon EBS Metrics (p. 535). **Note** To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period. Units: Count

Metric	Description
DiskReadBytes	Bytes read from all ephemeral disks available to the instance (if your instance uses Amazon EBS, see Amazon EBS Metrics (p. 535).)
	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.
	Units: Bytes
DiskWriteBytes	Bytes written to all ephemeral disks available to the instance (if your instance uses Amazon EBS, see Amazon EBS Metrics (p. 535).)
	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.
	Units: Bytes
NetworkIn	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.
	Units: Bytes
NetworkOut	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.
	Units: Bytes
StatusCheckFailed	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.
	Note Status check metrics are available at 1 minute frequency. For a newly launched instance, status check metric data will only be available after the instance has completed the initialization state. Status check metrics will become available within a few minutes of being in the running state.
	Units: Count
StatusCheckFailed_Instance	Reports whether the instance has passed the EC2 instance status check in the last minute. Values for this metric are either 0 (zero) or 1 (one.) A zero indicates that the status check passed. A one indicates a status check failure.
	Note Status check metrics are available at 1 minute frequency. For a newly launched instance, status check metric data will only be available after the instance has completed the initialization state. Status check metrics will become available within a few minutes of being in the running state.
	Units: Count

Metric	Description
StatusCheckFailed_System	Reports whether the instance has passed the EC2 system status check in the last minute. Values for this metric are either 0 (zero) or 1 (one.) A zero indicates that the status check passed. A one indicates a status check failure.
	Note Status check metrics are available at 1 minute frequency. For a newly launched instance, status check metric data will only be available after the instance has completed the initialization state. Status check metrics will become available within a few minutes of being in the running state.
	Units: Count

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 *AutoScalingGroup* is a collection of instances you define if you're using the Auto Scaling service. This dimension is available only for EC2 metrics when the instances are in such an AutoScalingGroup. Available for instances with Detailed or Basic Monitoring enabled.
ImageId	This dimension filters the data you request for all instances running this EC2 Amazon Machine Image (AMI). Available for instances with Detailed Monitoring enabled.
InstanceId	This dimension filters the data you request for the identified instance only. This helps you pinpoint an exact instance from which to monitor data. Available for instances with Detailed Monitoring enabled.
InstanceType	This dimension filters the data you request for all instances running with this specified instance type. This helps you categorize your data by the type of instance running. For example, you might compare data from an m1.small instance and an m1.large instance to determine which has the better business value for your application. Available for instances with Detailed Monitoring enabled.

For more information about using the GetMetricStatistics action, see GetMetricStatistics in the *Amazon CloudWatch API Reference*.

AWS Management Console

To view available metrics by category

You can view metrics by category. Metrics are grouped first by Namespace, and then by the various Dimension combinations within each Namespace. For example, you can view all EC2 metrics, or EC2 metrics grouped by instance ID, instance type, image (AMI) ID, or Auto Scaling Group.

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

3. In the navigation pane, click **Metrics**.
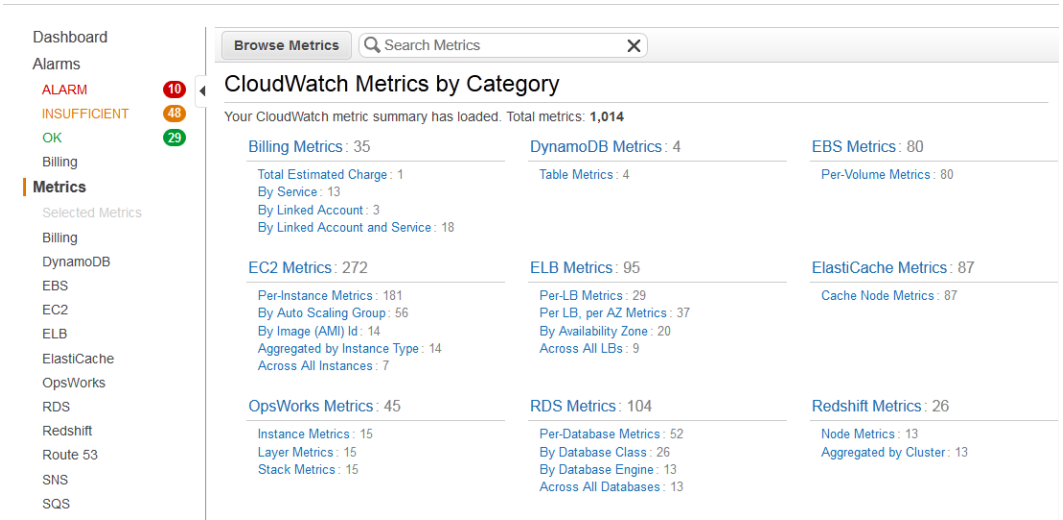
4. In the **CloudWatch Metrics by Category** pane, under **EC2 Metrics**, select **Per-Instance Metrics**, and then in the upper pane, scroll down to view the full list of metrics.

Command Line Interface

To list available metrics across multiple Amazon EC2 instances

Enter the `list-metrics` command and specify the AWS/EC2 namespace to limit the results to Amazon
EC2. For more information about the `list-metrics` command, see list-metrics in the *AWS Command
Line Interface Reference*.

```
$ aws cloudwatch list-metrics --namespace AWS/EC2
```

CloudWatch returns the following (partial listing):

```
{
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceType",
                "Value": "t1.micro"
            }
        ],
        "MetricName": "CPUUtilization"
    },
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-570e5a28"
            }
        ],
        "MetricName": "DiskWriteOps"
    },
```

```
        {
             "Namespace": "AWS/EC2",
             "Dimensions": [
                 {
                      "Name": "InstanceType",
                      "Value": "t1.micro"
                 }
             ],
             "MetricName": "NetworkOut"
        },
        {
             "Namespace": "AWS/EC2",
             "Dimensions": [
                 {
                      "Name": "ImageId",
                      "Value": "ami-6cb90605"
                 }
             ],
             "MetricName": "CPUUtilization"
        },
        {
             "Namespace": "AWS/EC2",
             "Dimensions": [
                 {
                      "Name": "ImageId",
                      "Value": "ami-6cb90605"
                 }
             ],
             "MetricName": "NetworkIn"
        },
        {
             "Namespace": "AWS/EC2",
             "Dimensions": [
                 {
                      "Name": "InstanceType",
                      "Value": "t1.micro"
                 }
             ],
             "MetricName": "DiskReadBytes"
        },
        {
             "Namespace": "AWS/EC2",
             "Dimensions": [
                 {
                      "Name": "InstanceId",
                      "Value": "i-570e5a28"
                 }
             ],
             "MetricName": "StatusCheckFailed_System"
        },
        {
             "Namespace": "AWS/EC2",
             "Dimensions": [
                 {
                      "Name": "InstanceId",
                      "Value": "i-570e5a28"
                 }
             ],
```

```
                "MetricName": "NetworkOut"
        },
        {

            "Namespace": "AWS/EC2",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "MetricName": "DiskWriteBytes"
        }
    ]
}
```

Get Statistics for Metrics

This set of scenarios shows you how you can use the AWS Management Console, the `get-metric-statistics` command, or the `GetMetricStatistics` API to get a variety of statistics.

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

Contents

- Get Statistics for a Specific EC2 Instance (p. 364)
- Aggregating Statistics Across Instances (p. 368)
- Get Statistics Aggregated by Auto Scaling Group (p. 373)
- Get Statistics Aggregated by Image (AMI) ID (p. 376)

Get Statistics for a Specific EC2 Instance

The following scenario walks you through how to use the AWS Management Console or the `get-metric-statistics` command to determine the maximum CPU utilization of a specific EC2 instance.

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

For this example, we assume that you have an EC2 instance ID. You can get an active EC2 instance ID through the AWS Management Console or with the `describe-instances` command.

AWS Management Console

To display the average CPU utilization for a specific instance

1. Open the Amazon CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see Regions and Endpoints.
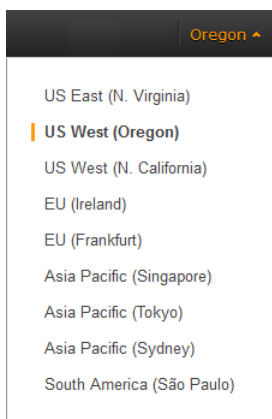
3. In the navigation pane, click **Metrics**.

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

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

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

   A graph showing average `CPUUtilization` for a single instance appears in the details pane.
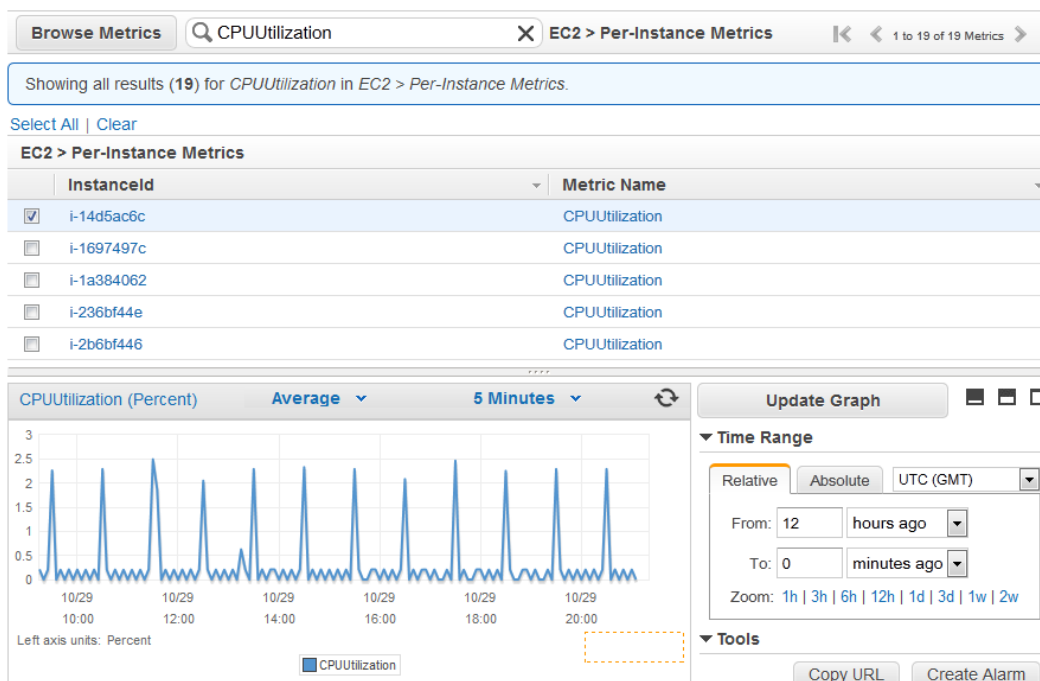
6. To change the **Statistic**, e.g., Average, for the metric, choose a different value from the pop-up list.
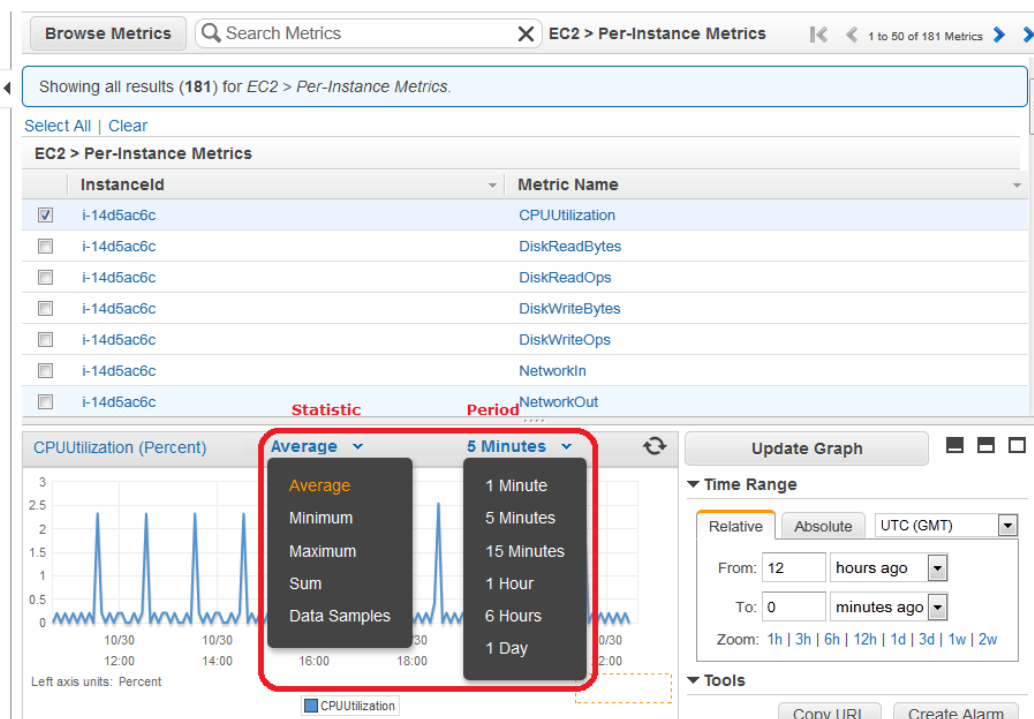
7. To change the **Period**, e.g., 5 Minutes, to view data in more granular detail, choose a different value from the pop-up list.

Command Line Interface

To get the CPU utilization per EC2 instance

Enter the `get-metric-statistics` command with the following parameters. For more information about the `get-metric-statistics` command, see get-metric-statistics in the *AWS Command Line Interface Reference*.

```
$ aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-
time 2014-02-18T23:18:00 --end-time 2014-02-19T23:18:00 --period 3600 --namespace
 AWS/EC2 --statistics Maximum --dimensions Name=InstanceId,Value=<your-instance-
id>
```

The AWS CLI returns the following:

```
{
    "Datapoints": [
        {
            "Timestamp": "2014-02-19T00:18:00Z",
            "Maximum": 0.33000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T03:18:00Z",
            "Maximum": 99.670000000000002,
            "Unit": "Percent"
```

```
        },
        {
            "Timestamp": "2014-02-19T07:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T12:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T02:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T01:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T17:18:00Z",
            "Maximum": 3.3900000000000001,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T13:18:00Z",
            "Maximum": 0.33000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-18T23:18:00Z",
            "Maximum": 0.67000000000000004,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T06:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T11:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T10:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T19:18:00Z",
            "Maximum": 8.0,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-19T15:18:00Z",
```

```
                     "Maximum": 0.34000000000000002,
                     "Unit": "Percent"
             },
             {
                     "Timestamp": "2014-02-19T14:18:00Z",
                     "Maximum": 0.34000000000000002,
                     "Unit": "Percent"
             },
             {
                     "Timestamp": "2014-02-19T16:18:00Z",
                     "Maximum": 0.34000000000000002,
                     "Unit": "Percent"
             },
             {
                     "Timestamp": "2014-02-19T09:18:00Z",
                     "Maximum": 0.34000000000000002,
                     "Unit": "Percent"
             },
             {
                     "Timestamp": "2014-02-19T04:18:00Z",
                     "Maximum": 2.0,
                     "Unit": "Percent"
             },
             {
                     "Timestamp": "2014-02-19T08:18:00Z",
                     "Maximum": 0.68000000000000005,
                     "Unit": "Percent"
             },
             {
                     "Timestamp": "2014-02-19T05:18:00Z",
                     "Maximum": 0.33000000000000002,
                     "Unit": "Percent"
             },
             {
                     "Timestamp": "2014-02-19T18:18:00Z",
                     "Maximum": 6.6699999999999999,
                     "Unit": "Percent"
             }
     ],
     "Label": "CPUUtilization"
}
```

The returned statistics are six-minute values for the requested two-day time interval. Each value represents the maximum CPU utilization percentage for a single EC2 instance.

Aggregating Statistics Across Instances

Aggregate statistics are available for the instances that have detailed monitoring enabled. Instances that use basic monitoring are not included in the aggregates. In addition, Amazon CloudWatch does not aggregate data across Regions. Therefore, metrics are completely separate between Regions. Before you can get statistics aggregated across instances, you must enable detailed monitoring (at an additional charge), which provides data in 1-minute periods. This scenario shows you how to use detailed monitoring with either the AWS Management Console, the GetMetricStatistics API, or the get-metric-statistics command to get the average CPU usage for your EC2 instances. Because no dimension is specified, CloudWatch returns statistics for all dimensions in the AWS/EC2 namespace. To get statistics for other metrics, see Amazon CloudWatch Namespaces, Dimensions, and Metrics Reference.

Important
This technique for retrieving all dimensions across an AWS namespace does not work for custom namespaces that you publish to Amazon CloudWatch. With custom namespaces, you must specify the complete set of dimensions that are associated with any given data point to retrieve statistics that include the data point.

AWS Management Console

To display average CPU utilization for your Amazon EC2 instances

1. Open the Amazon CloudWatch console at https://console.aws.amazon.com/cloudwatch/.

2. If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see Regions and Endpoints.
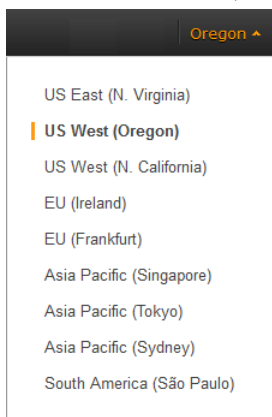
3. In the navigation pane, click **Metrics**.

4. In the **CloudWatch Metrics by Category** pane, under **EC2 Metrics**, select **Across All Instances**.

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

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

   A graph showing CPUUtilization for your EC2 instances is displayed in the details pane.
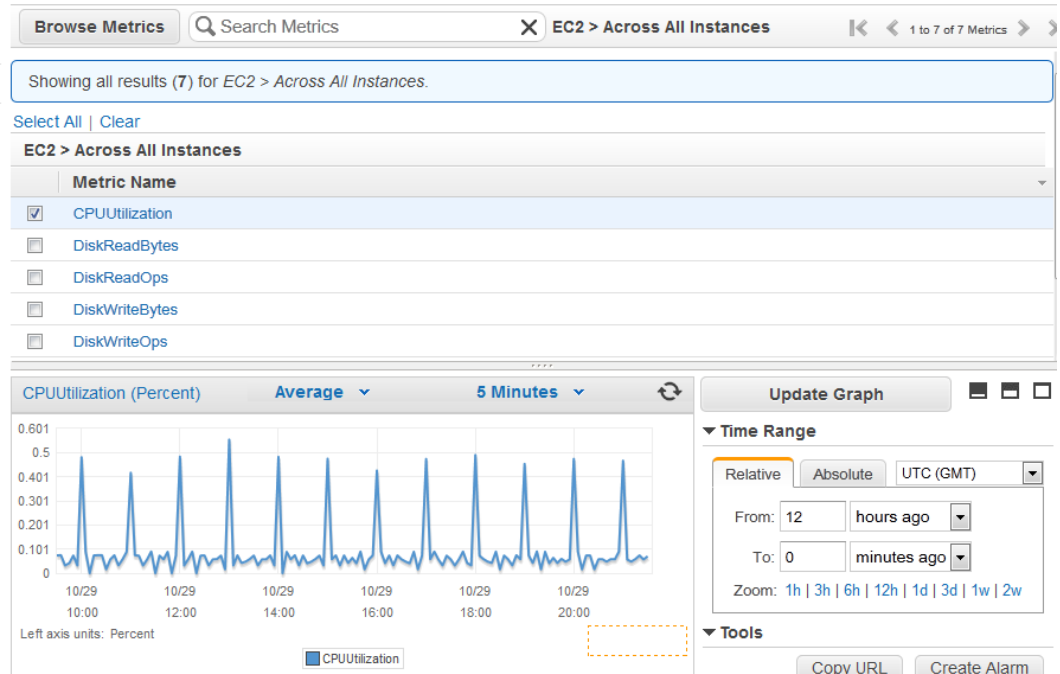
6.   To change the **Statistic**, e.g., Average, for the metric, choose a different value from the pop-up list.
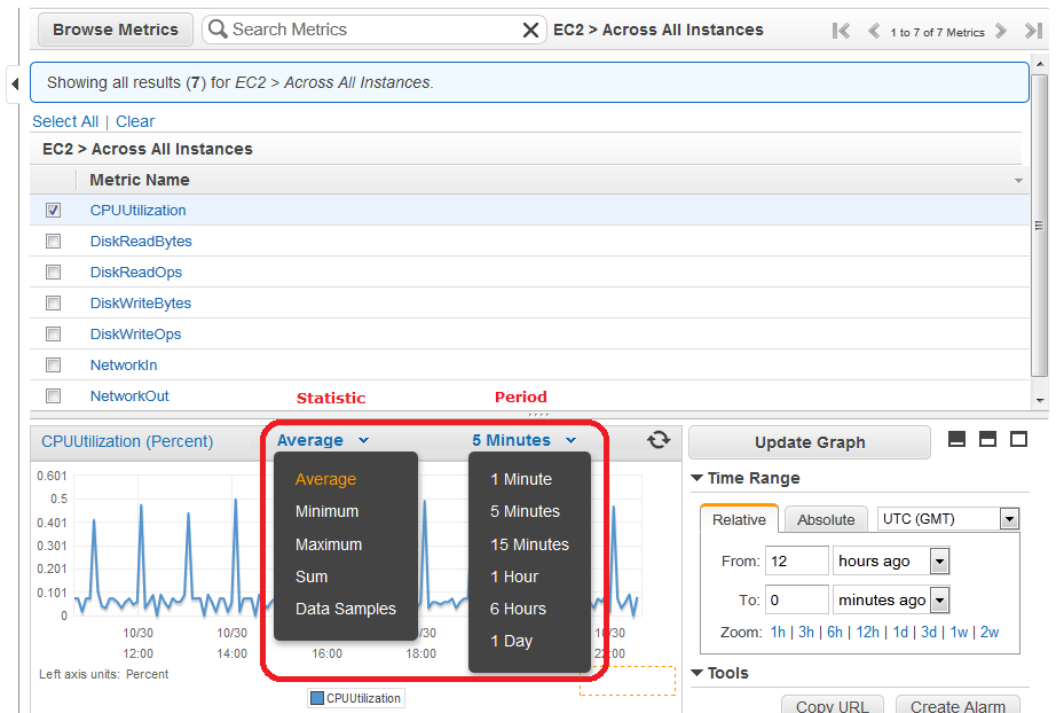
7.   To change the **Period**, e.g., 5 Minutes, to view data in more granular detail, choose a different value from the pop-up list.

Command Line Interface

To get average CPU utilization across your Amazon EC2 instances

Enter the `get-metric-statistics` command with the following parameters. For more information about the `get-metric-statistics` command, see get-metric-statistics in the *AWS Command Line Interface Reference*.

```
$ aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-
time 2014-02-11T23:18:00 --end-time 2014-02-12T23:18:00 --period 3600 --namespace
 AWS/EC2 --statistics "Average" "SampleCount"
```

The AWS CLI returns the following:

```
{
    "Datapoints": [
        {
            "SampleCount": 238.0,
            "Timestamp": "2014-02-12T07:18:00Z",
            "Average": 0.038235294117647062,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2014-02-12T09:18:00Z",
            "Average": 0.16670833333333332,
            "Unit": "Percent"
        },
        {
            "SampleCount": 238.0,
            "Timestamp": "2014-02-11T23:18:00Z",
            "Average": 0.041596638655462197,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2014-02-12T16:18:00Z",
            "Average": 0.039458333333333345,
            "Unit": "Percent"
        },
        {
            "SampleCount": 239.0,
            "Timestamp": "2014-02-12T21:18:00Z",
            "Average": 0.041255230125523033,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2014-02-12T01:18:00Z",
            "Average": 0.044583333333333336,
            "Unit": "Percent"
        },
        {
            "SampleCount": 239.0,
            "Timestamp": "2014-02-12T18:18:00Z",
            "Average": 0.043054393305439344,
            "Unit": "Percent"
```

```
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2014-02-12T13:18:00Z",
            "Average": 0.039458333333333345,
            "Unit": "Percent"
        },
        {
            "SampleCount": 238.0,
            "Timestamp": "2014-02-12T15:18:00Z",
            "Average": 0.041260504201680689,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2014-02-12T19:18:00Z",
            "Average": 0.037666666666666668,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2014-02-12T06:18:00Z",
            "Average": 0.037541666666666675,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2014-02-12T20:18:00Z",
            "Average": 0.039333333333333338,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2014-02-12T08:18:00Z",
            "Average": 0.039250000000000014,
            "Unit": "Percent"
        },
        {
            "SampleCount": 239.0,
            "Timestamp": "2014-02-12T03:18:00Z",
            "Average": 0.037740585774058588,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2014-02-12T11:18:00Z",
            "Average": 0.039500000000000007,
            "Unit": "Percent"
        },
        {
            "SampleCount": 238.0,
            "Timestamp": "2014-02-12T02:18:00Z",
            "Average": 0.039789915966386563,
            "Unit": "Percent"
        },
        {
            "SampleCount": 238.0,
            "Timestamp": "2014-02-12T22:18:00Z",
```

```
                "Average": 0.039705882352941181,
                "Unit": "Percent"
            },
            {

                "SampleCount": 240.0,
                "Timestamp": "2014-02-12T14:18:00Z",
                "Average": 0.082458333333333328,
                "Unit": "Percent"
            },
            {

                "SampleCount": 240.0,
                "Timestamp": "2014-02-12T05:18:00Z",
                "Average": 0.04287500000000001,
                "Unit": "Percent"
            },
            {

                "SampleCount": 240.0,
                "Timestamp": "2014-02-12T17:18:00Z",
                "Average": 0.039458333333333345,
                "Unit": "Percent"
            },
            {

                "SampleCount": 240.0,
                "Timestamp": "2014-02-12T10:18:00Z",
                "Average": 0.083416666666666667,
                "Unit": "Percent"
            },
            {

                "SampleCount": 236.0,
                "Timestamp": "2014-02-12T00:18:00Z",
                "Average": 0.036567796610169498,
                "Unit": "Percent"
            },
            {

                "SampleCount": 240.0,
                "Timestamp": "2014-02-12T12:18:00Z",
                "Average": 0.039541666666666676,
                "Unit": "Percent"
            },
            {

                "SampleCount": 240.0,
                "Timestamp": "2014-02-12T04:18:00Z",
                "Average": 0.043000000000000003,
                "Unit": "Percent"
            }
        ],
        "Label": "CPUUtilization"
}
```

Get Statistics Aggregated by Auto Scaling Group

This scenario shows you how to use the AWS Management Console, the `get-metric-statistics` command, or the `GetMetricStatistics` API with the *DiskWriteBytes* metric to retrieve the total bytes written to disk for one Auto Scaling group. The total is computed for one-minute periods for a 24-hour interval across all EC2 instances in the specified *AutoScalingGroupName*.

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

We assume for this example that an EC2 application is running and has an Auto Scaling group named `test-group-1`.

AWS Management Console

To display total DiskWriteBytes for an Auto-Scaled EC2 application

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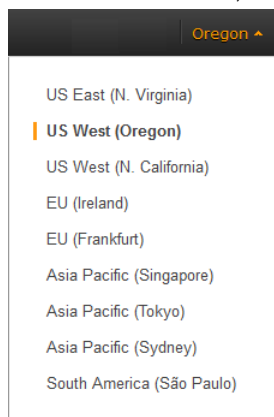

3. In the navigation pane, click **Metrics**.
4. In the **CloudWatch Metrics by Category** pane, under **EC2 Metrics**, select **By Auto Scaling Group**.

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

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

   A graph showing `DiskWriteBytes` for all EC2 instances appears in the details pane.
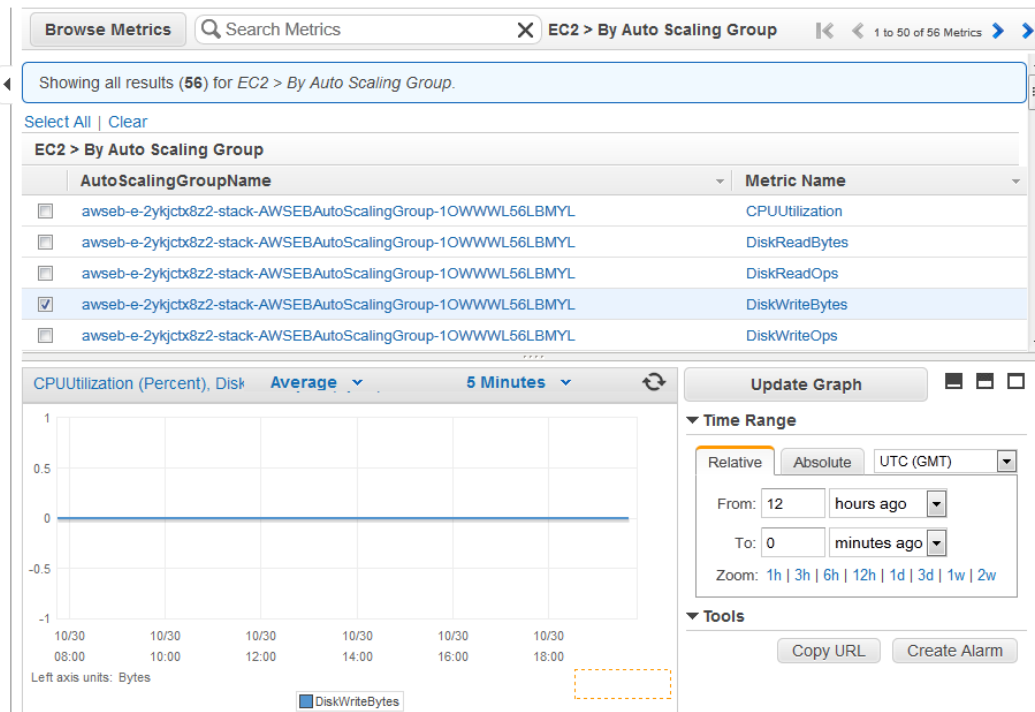
6. To change the **Statistic**, e.g., Average, for the metric, choose a different value from the pop-up list.
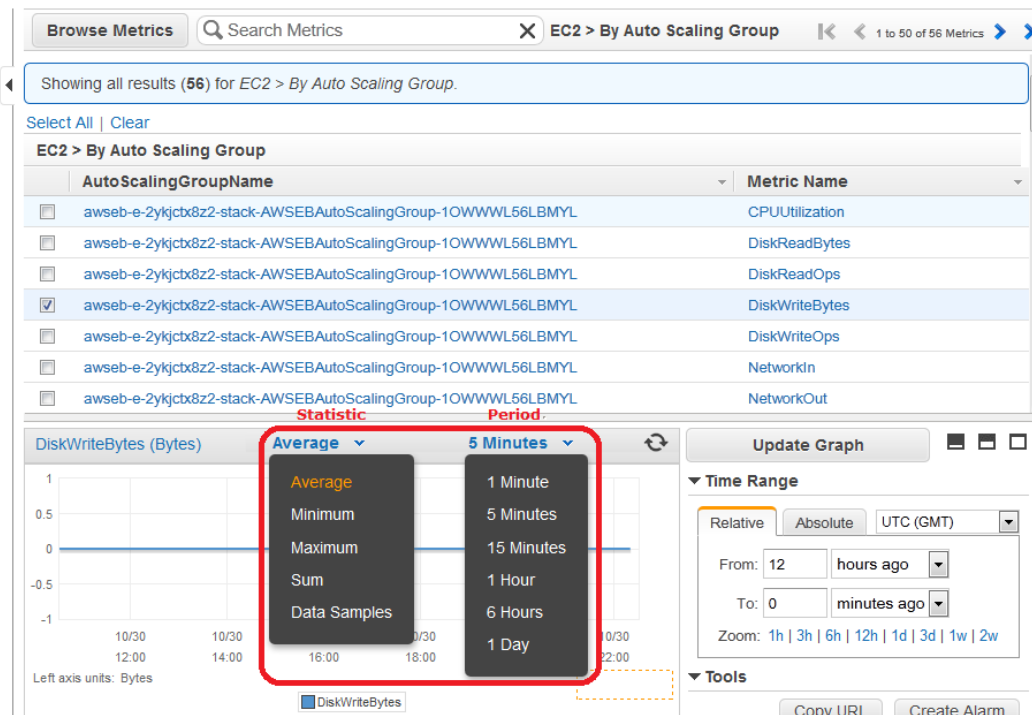
7. To change the **Period**, e.g., 5 Minutes, to view data in more granular detail, choose a different value from the pop-up list.

Command Line Interface

To get total DiskWriteBytes for an auto-scaled EC2 application

Enter the `get-metric-statistics` command with the following parameters. For more information about the `get-metric-statistics` command, see get-metric-statistics in the *AWS Command Line Interface Reference*.

```
$ aws cloudwatch get-metric-statistics --metric-name DiskWriteBytes --start-
time 2014-02-16T23:18:00 --end-time 2014-02-18T23:18:00 --period 360 --namespace
 AWS/EC2 --statistics "Sum" "SampleCount" --dimensions Name=AutoScalingGroup
Name,Value=test-group-1
```

The AWS CLI returns the following:

```
{
    "Datapoints": [
        {
            "SampleCount": 18.0,
            "Timestamp": "2014-02-19T21:36:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "SampleCount": 5.0,
```

```
            "Timestamp": "2014-02-19T21:42:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        }
    ],
    "Label": "DiskWriteBytes"
}
```

Get Statistics Aggregated by Image (AMI) ID

This scenario shows you how to use the AWS Management Console, the `get-metric-statistics` command, or the `GetMetricStatistics` API to determine average CPU utilization for all instances that match a given image ID. The average is over 60-second time intervals for a one-day period.

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

In this scenario, the EC2 instances are running an image ID of `ami-c5e40dac`.

AWS Management Console

To display the average CPU utilization for an image ID

1. Open the Amazon CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see Regions and Endpoints.
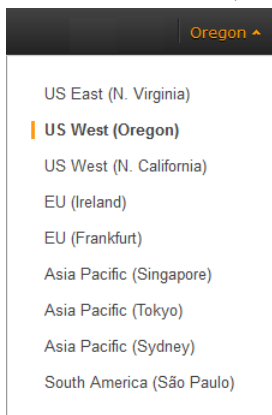
3. In the navigation pane, click **Metrics**.
4. In the **CloudWatch Metrics by Category** pane, under **EC2 Metrics**, select **By Image (AMI) Id**.

    The metrics available for image IDs appear in the upper pane.
5. Select a row that contains **CPUUtilization** and an image ID.

    A graph showing average `CPUUtilization` for all EC2 instances based on the `ami-c5e40dac` image ID appears in the details pane.
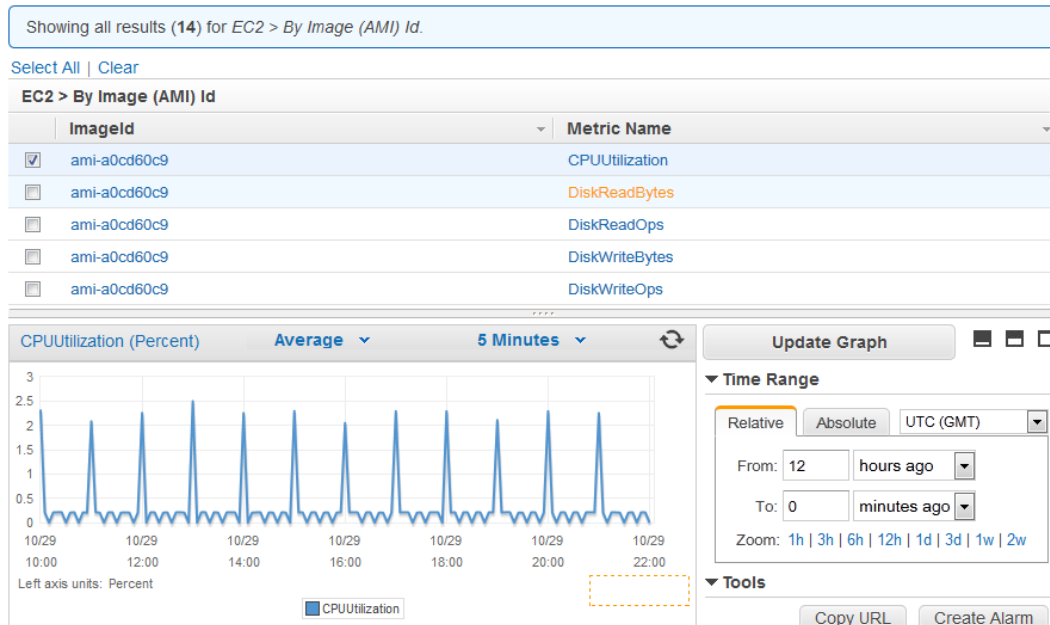
6. To change the **Statistic**, e.g., Average, for the metric, choose a different value from the pop-up list.
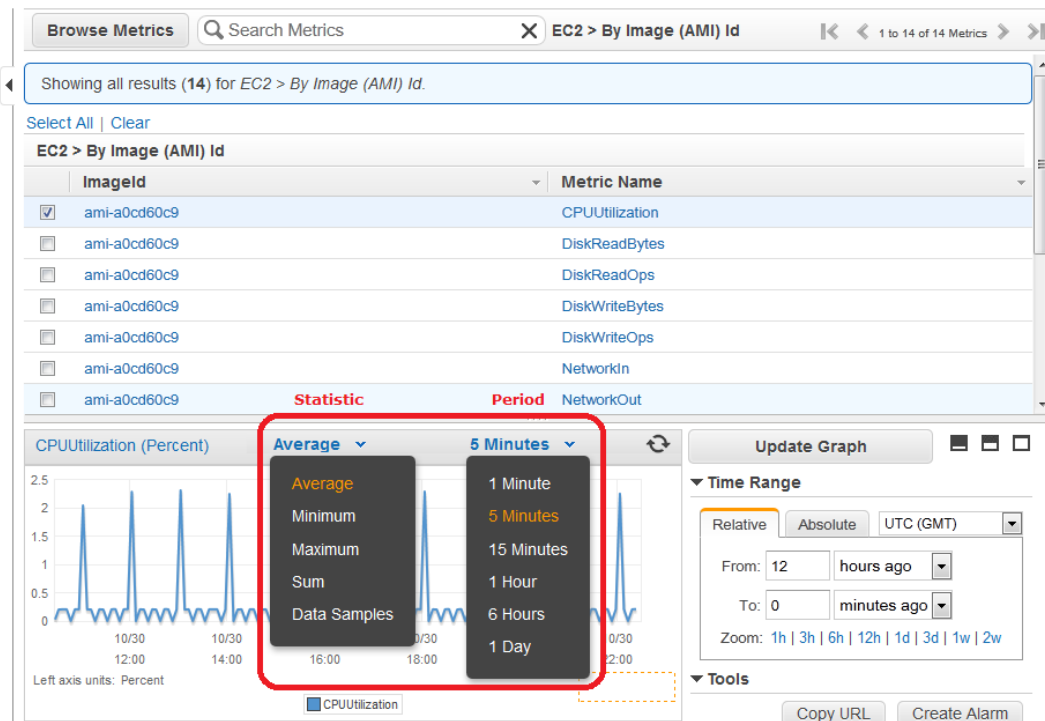
7. To change the **Period**, e.g., 5 Minutes, to view data in more granular detail, choose a different value from the pop-up list.

Command Line Interface

To get the average CPU utilization for an image ID

Enter the `get-metric-statistics` command as in the following example. For more information about the `get-metric-statistics` command, see get-metric-statistics in the *AWS Command Line Interface Reference*.

```
$ aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-
time 2014-02-10T00:00:00 --end-time 2014-02-11T00:00:00 --period 3600 --statist
ics Average --namespace AWS/EC2 --dimensions Name="ImageId",Value=ami-3c47a355"
```

The AWS CLI returns the following:

```
{
    "Datapoints": [
        {
            "Timestamp": "2014-02-10T07:00:00Z",
            "Average": 0.041000000000000009,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T14:00:00Z",
            "Average": 0.079579831932773085,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T06:00:00Z",
            "Average": 0.036000000000000011,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T13:00:00Z",
            "Average": 0.037625000000000013,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T18:00:00Z",
            "Average": 0.042750000000000003,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T21:00:00Z",
            "Average": 0.039705882352941188,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T20:00:00Z",
            "Average": 0.039375000000000007,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T02:00:00Z",
            "Average": 0.041041666666666671,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-02-10T01:00:00Z",
            "Average": 0.041083333333333354,
            "Unit": "Percent"
        },
```

```
                {
                    "Timestamp": "2014-02-10T23:00:00Z",
                    "Average": 0.038016877637130804,
                    "Unit": "Percent"
                },
                {
                    "Timestamp": "2014-02-10T15:00:00Z",
                    "Average": 0.037666666666666668,
                    "Unit": "Percent"
                },
                {
                    "Timestamp": "2014-02-10T12:00:00Z",
                    "Average": 0.039291666666666676,
                    "Unit": "Percent"
                },
                {
                    "Timestamp": "2014-02-10T03:00:00Z",
                    "Average": 0.036000000000000004,
                    "Unit": "Percent"
                },
                {
                    "Timestamp": "2014-02-10T04:00:00Z",
                    "Average": 0.042666666666666672,
                    "Unit": "Percent"
                },
                {
                    "Timestamp": "2014-02-10T19:00:00Z",
                    "Average": 0.038305084745762719,
                    "Unit": "Percent"
                },
                {
                    "Timestamp": "2014-02-10T22:00:00Z",
                    "Average": 0.039291666666666676,
                    "Unit": "Percent"
                },
                {
                    "Timestamp": "2014-02-10T09:00:00Z",
                    "Average": 0.17126050420168065,
                    "Unit": "Percent"
                },
                {
                    "Timestamp": "2014-02-10T08:00:00Z",
                    "Average": 0.041166666666666678,
                    "Unit": "Percent"
                },
                {
                    "Timestamp": "2014-02-10T11:00:00Z",
                    "Average": 0.082374999999999962,
                    "Unit": "Percent"
                },
                {
                    "Timestamp": "2014-02-10T17:00:00Z",
                    "Average": 0.037625000000000013,
                    "Unit": "Percent"
                },
                {
                    "Timestamp": "2014-02-10T10:00:00Z",
                    "Average": 0.039458333333333345,
```

```
                    "Unit": "Percent"
                },
                {

                    "Timestamp": "2014-02-10T05:00:00Z",
                    "Average": 0.039250000000000007,
                    "Unit": "Percent"
                },
                {

                    "Timestamp": "2014-02-10T00:00:00Z",
                    "Average": 0.037625000000000013,
                    "Unit": "Percent"
                },
                {

                    "Timestamp": "2014-02-10T16:00:00Z",
                    "Average": 0.041512605042016815,
                    "Unit": "Percent"
                }
        ],
        "Label": "CPUUtilization"
}
```

The operation returns statistics that are one-minute values for the one-day interval. Each value represents an average CPU utilization percentage for EC2 instances running the specified machine image.

Graphing Metrics

After you launch an instance, you can go to the Amazon EC2 console and view the instance's monitoring graphs. They're displayed when you select the instance on the **Instances** page in the EC2 Dashboard. A **Monitoring** tab is displayed next to the instance's **Description** tab. The following graphs are available:

- Average CPU Utilization (Percent)
- Average Disk Reads (Bytes)
- Average Disk Writes (Bytes)
- Maximum Network In (Bytes)
- Maximum Network Out (Bytes)
- Summary Disk Read Operations (Count)
- Summary Disk Write Operations (Count)
- Summary Status (Any)
- Summary Status Instance (Count)
- Summary Status System (Count)

You can also use the CloudWatch console to graph metric data generated by Amazon EC2 and other AWS services to make it easier to see what's going on. You can use the following procedures to graph metrics in CloudWatch.

Contents

Graph a Metric

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

To graph a metric

1. Open the Amazon CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see Regions and Endpoints in the *Amazon Web Services General Reference*.
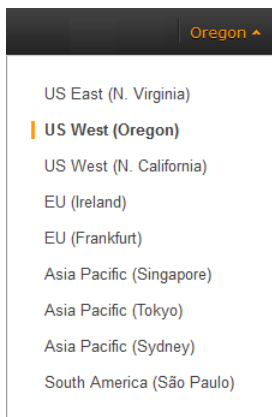
3. In the navigation pane, click **Metrics**.
4. In the **CloudWatch Metrics by Category** pane, use the **Search Metrics** box and categories to find a metric by metric name, AWS resource, or other metadata.
5. Use the scroll bar and next and previous arrows above the metrics list to page through the full list of metrics
6. Select the metric to view, for example, CPUUtilization. A graph appears in the details pane.
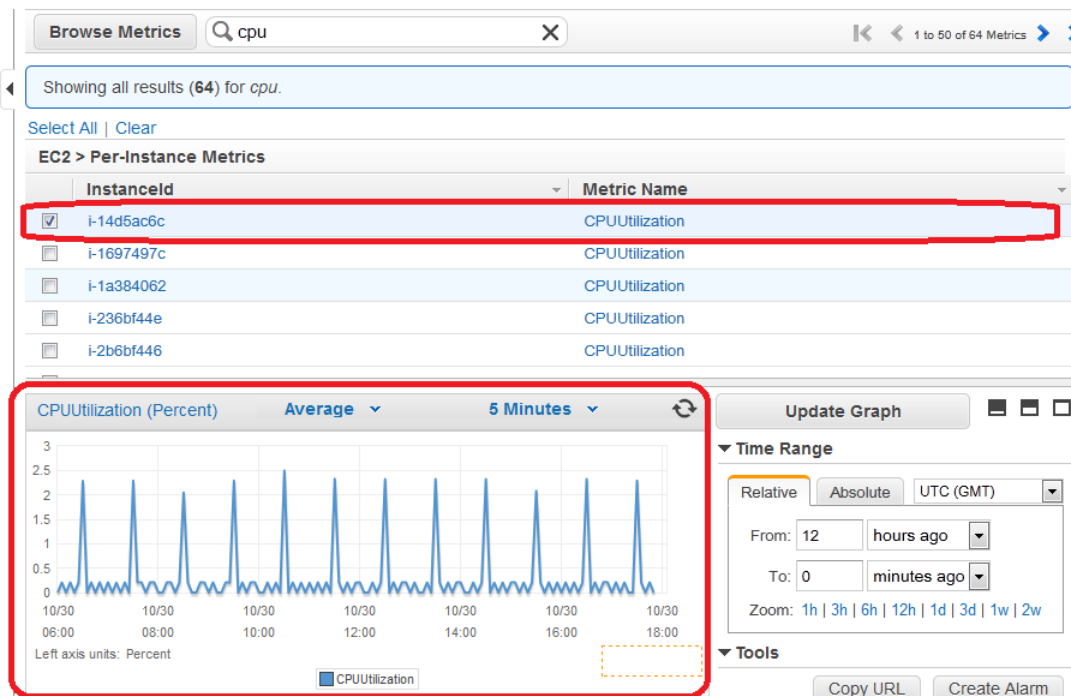
7. To save this graph and access it later, in the details pane, under **Tools**, click **Copy URL**, and then in the **Copy Graph URL** dialog box, select the URL and paste it into your browser.

Graph a Metric Across Resources

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

To graph a metric across resources

1. Open the Amazon CloudWatch console at https://console.aws.amazon.com/cloudwatch/.

2. If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see Regions and Endpoints.
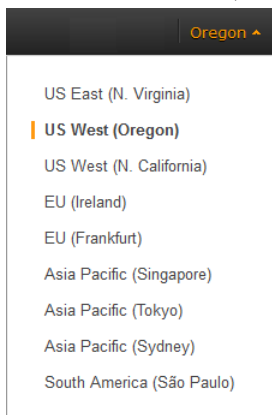
3. In the navigation pane, click **Metrics**.
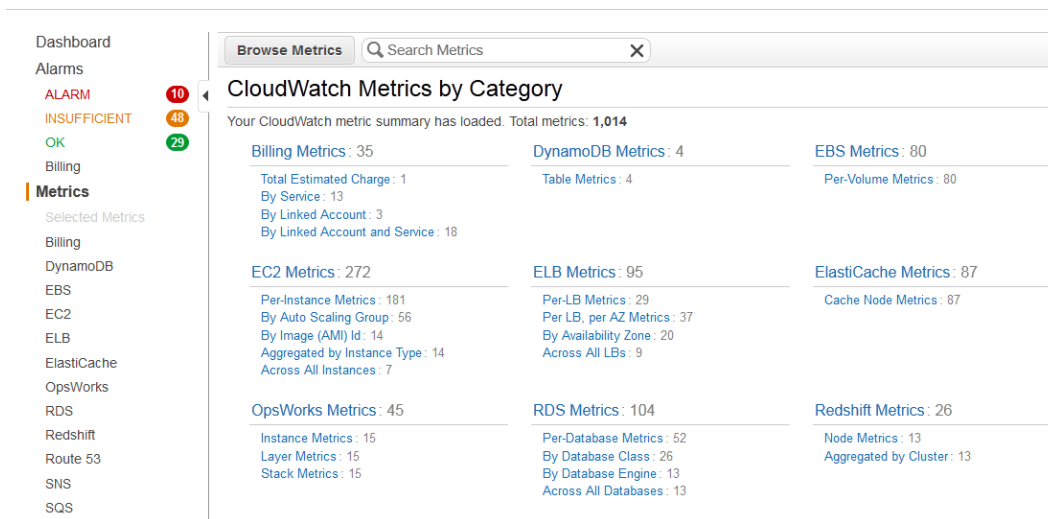
4. In the **CloudWatch Metrics by Category** pane, select a metric category. For example, under **EC2 Metrics**, select **Per-Instance Metrics**.
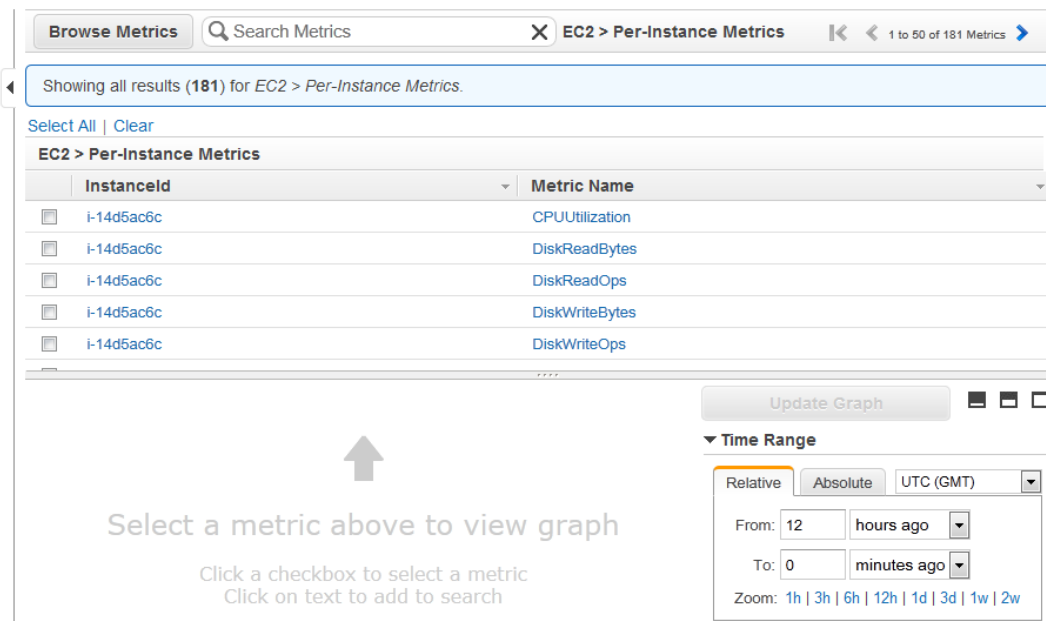
5. In the metric list, in the **Metric Name** column, click a metric. For example **CPUUtilization**.

6. At the top of the metric list, click **Select All**.

   The graph shows all data for all occurrences of the selected metric. In the example below, CPUUtil-
   ization for all Amazon EC2 instances is shown.
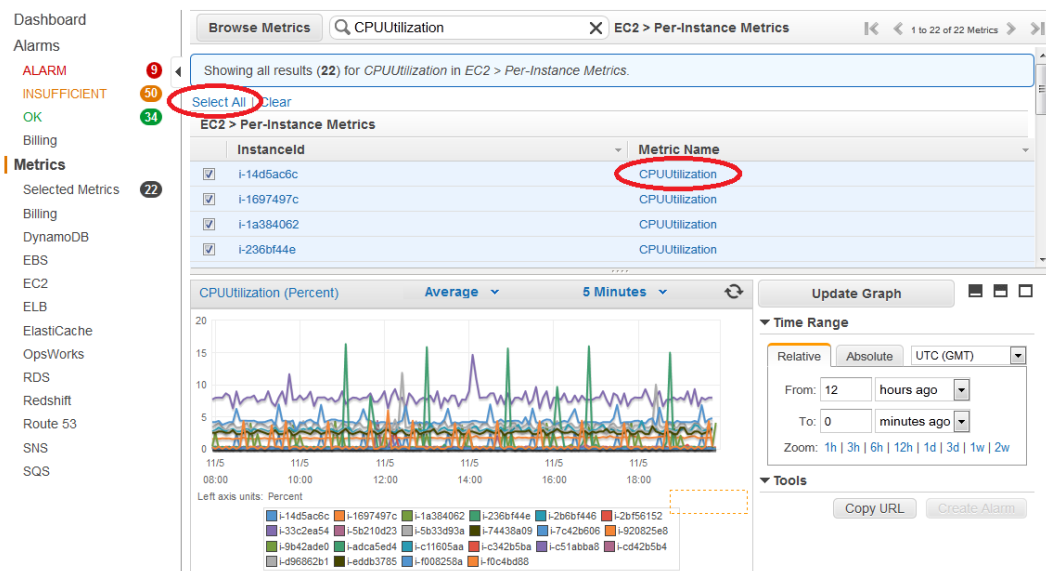
7. To save this graph and access it later, in the details pane, under **Tools**, click **Copy URL**, and then
   in the **Copy Graph URL** dialog box, select the URL and paste it into your browser.

Create a CloudWatch Alarm

You can create an Amazon CloudWatch alarm that monitors any one of your Amazon EC2 instance's CloudWatch metrics. CloudWatch will automatically send you a notification when the metric reaches a threshold you specify. You can create a CloudWatch alarm on the Amazon EC2 console of the AWS Management Console, or you can use the CloudWatch console and configure more advanced options.

Contents

- Send Email Based on CPU Usage Alarm (p. 384)
- Send Email Based on Load Balancer Alarm (p. 386)
- Send Email Based on Storage Throughput Alarm (p. 388)

Send Email Based on CPU Usage Alarm

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

In this scenario, you configure the alarm to change to the ALARM state when the average CPU use of an EC2 instance exceeds 70 percent for two consecutive five-minute periods.

AWS Management Console

To create an alarm that sends email based on CPU usage

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

Oregon ▲
US East (N. Virginia)
US West (Oregon)
US West (N. California)
EU (Ireland)
EU (Frankfurt)
Asia Pacific (Singapore)
Asia Pacific (Tokyo)
Asia Pacific (Sydney)
South America (São Paulo)
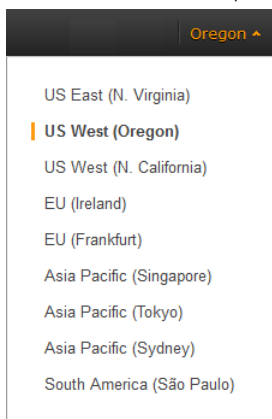
3. In the navigation pane, click **Alarms**.
4. Click **Create Alarm**, and then in **CloudWatch Metrics by Category**, select a metric category, for example, **EC2 Metrics**.
5. In the list of metrics, select a row that contains **CPUUtilization** for a specific instance ID.

   A graph showing average `CPUUtilization` for a single instance appears in the lower pane.
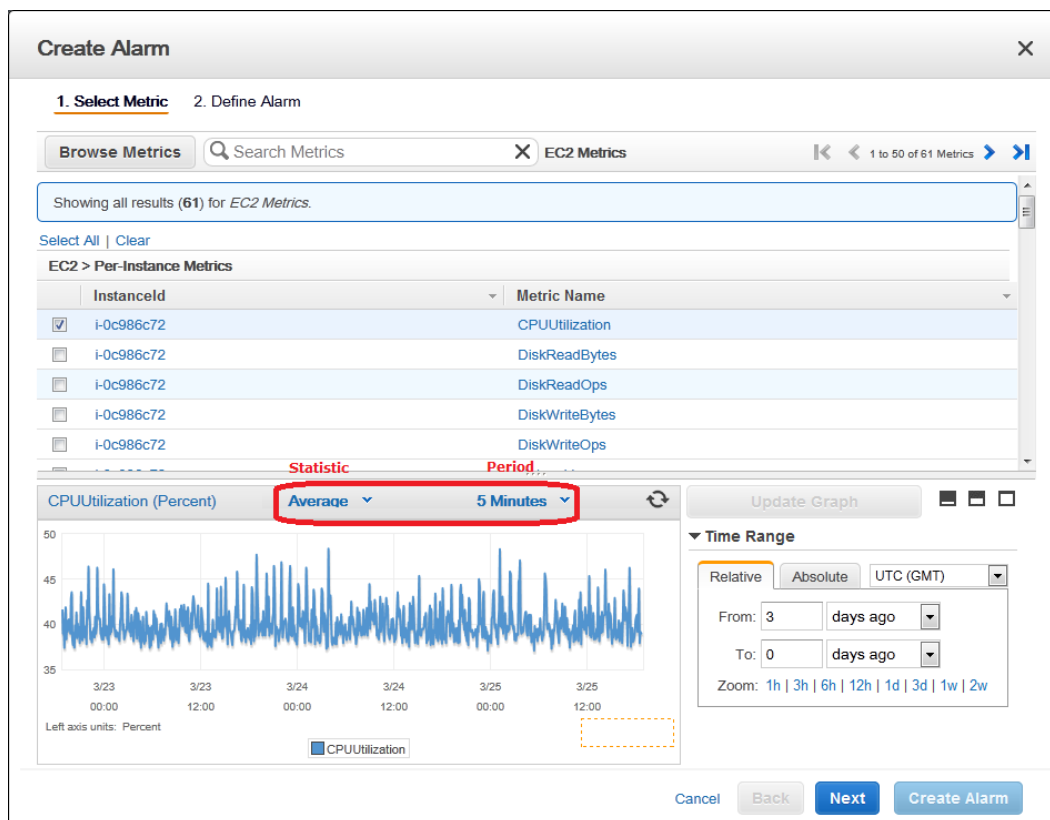
6. Select **Average** from the **Statistic** drop-down list.

7. Select a period from the **Period** drop-down list, for example: `5 minutes`.

8. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: `myHighCpuAlarm`.

9. In the **Description** field, enter a description of the alarm, for example: `CPU usage exceeds 70 percent.`

10. In the **is** drop-down list, select `>`.

11. In the box next to the **is** drop-down list, enter `70` and in the **for** field, enter `10`.

    A graphical representation of the threshold is shown under **Alarm Preview**.

12. Under **Actions**, in the **Whenever this alarm** drop-down list, select **State is ALARM**.

13. In the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.

14. To create a new Amazon SNS topic, select **New list**.

    In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: `myHigh-CpuAlarm`, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the `ALARM` state.

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

Command Line Interface

To send an Amazon Simple Notification Service email message when CPU utilization exceeds 70 percent

1. Set up an Amazon Simple Notification Service topic or retrieve the Topic Resource Name of the topic you intend to use. For help on setting up an Amazon Simple Notification Service topic, see Set Up Amazon Simple Notification Service.

2. Create an alarm with the `put-metric-alarm` command. For more information about the `put-metric-alarm` command, see put-metric-alarm in the *AWS Command Line Interface Reference*. Use the values from the following example, but replace the values for `InstanceID` and `alarm-actions` with your own values.

```
$ aws cloudwatch
              put-metric-alarm --alarm-name cpu-mon --alarm-description
"Alarm when CPU exceeds 70%" --metric-name CPUUtilization --namespace AWS/EC2
 --statistic Average --period 300
              --threshold 70 --comparison-operator GreaterThanThreshold  -
-dimensions Name=InstanceId,Value=i-12345678 --evaluation-periods 2 --alarm-
actions arn:aws:sns:us-east-1:111122223333:MyTopic --unit Percent
```

The AWS CLI returns to the command prompt if the command succeeds.

3. Test the alarm by forcing an alarm state change with the `set-alarm-state` command.

   a. Change the alarm state from `INSUFFICIENT_DATA` to `OK`:

   ```
   $ aws cloudwatch set-alarm-state  --alarm-name cpu-mon --state-reason
   "initializing" --state-value OK
   ```

   The AWS CLI returns to the command prompt if the command succeeds.

   b. Change the alarm state from `OK` to `ALARM`:

   ```
   $ aws cloudwatch set-alarm-state  --alarm-name cpu-mon --state-reason
   "initializing" --state-value ALARM
   ```

   The AWS CLI returns to the command prompt if the command succeeds.

   c. Check that an email has been received.

Send Email Based on Load Balancer Alarm

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

AWS Management Console

To create a load balancer alarm that sends email

1. Open the Amazon CloudWatch console at https://console.aws.amazon.com/cloudwatch/.

2. If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see Regions and Endpoints.
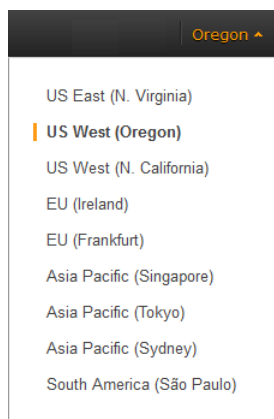
3. In the navigation pane, click **Alarms**.

4. Click **Create Alarm**, and then in the **CloudWatch Metrics by Category** pane, select a metric category, for example, **ELB Metrics**.

5. In the list of metrics, select a row that contains **Latency** for a specific load balancer.

   A graph showing average `Latency` for a single load balancer appears in the lower pane.
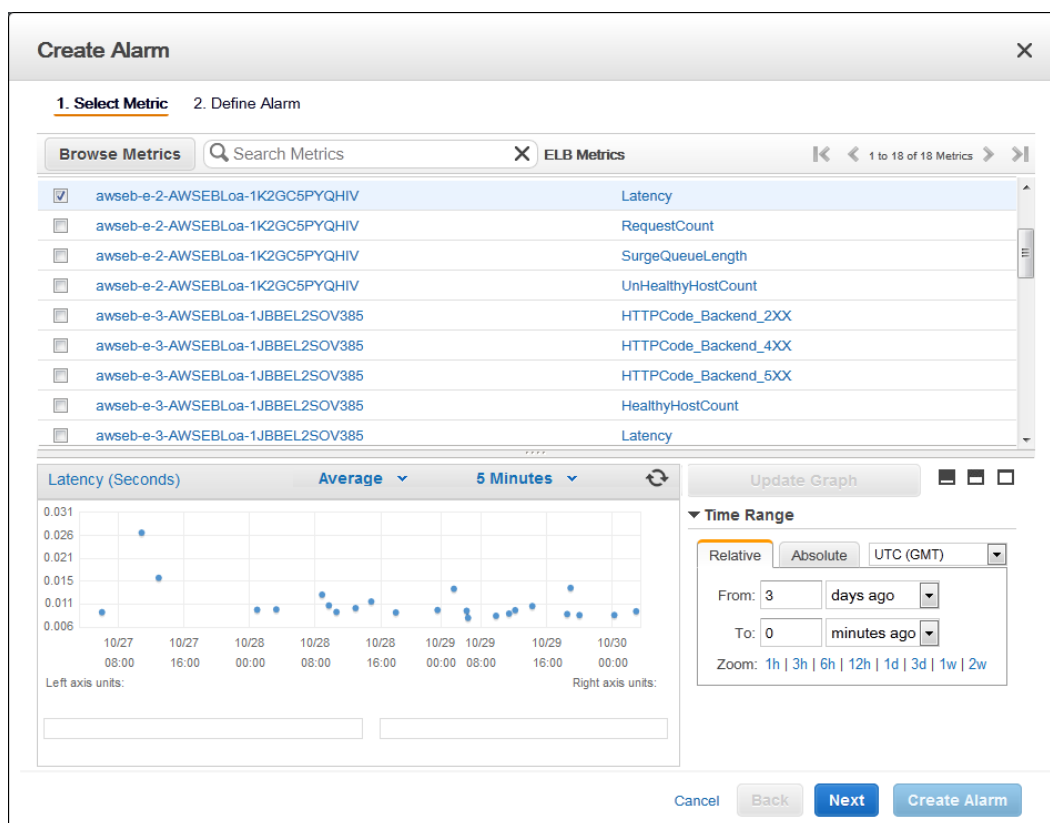
6. Select **Average** from the **Statistic** drop-down list.

7. Select **1 Minute** from the **Period** drop-down list.

8. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: `myHighCpuAlarm`.

9. In the **Description** field, enter a description of the alarm, for example: `Alarm when Latency exceeds 100ms.`

10. In the **is** drop-down list, select **>**.

11. In the box next to the **is** drop-down list, enter `0.1` and in the **for** field, enter `3`.

    A graphical representation of the threshold is shown under **Alarm Preview**.

12. Under **Actions**, in the **Whenever this alarm** drop-down list, select **State is ALARM**.

13. In the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.

14. To create a new Amazon SNS topic, select **New list**.

    In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: `myHigh-CpuAlarm`, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the `ALARM` state.

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

Command Line Interface

To send an Amazon Simple Notification Service email message when LoadBalancer Latency Exceeds 100 milliseconds

1. Create an Amazon Simple Notification Service topic. See instructions for creating an Amazon SNS topic in Set Up Amazon Simple Notification Service.

2. Use the `put-metric-alarm` command to create an alarm. For more information about the `put-metric-alarm` command, see put-metric-alarm in the *AWS Command Line Interface Reference*.

```
$ aws cloudwatch put-metric-alarm --alarm-name lb-mon --alarm-description
"Alarm when Latency exceeds 100ms" --metric-name Latency --namespace AWS/ELB
 --statistic Average --period 60 --threshold 100 --comparison-operator
GreaterThanThreshold --dimensions Name=LoadBalancerName,Value=my-server --
evaluation-periods 3 --alarm-actions arn:aws:sns:us-east-1:1234567890:my-
topic --unit Milliseconds
```

    The AWS CLI returns to the command prompt if the command succeeds.

3. Test the alarm.

    • Force an alarm state change to ALARM:

```
$ aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason "ini
tializing" --state OK
$ aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason "ini
tializing" --state ALARM
```

    The AWS CLI returns to the command prompt if the command succeeds.

    • Check that an email has been received.

Send Email Based on Storage Throughput Alarm

This scenario walks you through how to use the AWS Management Console or the command line interface to set up an Amazon Simple Notification Service notification and to configure an alarm that sends email when EBS exceeds 100 MB throughput.

AWS Management Console

To create a storage throughput alarm that sends email

1. Open the Amazon CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region that meets your needs. For more information, see Regions and Endpoints.
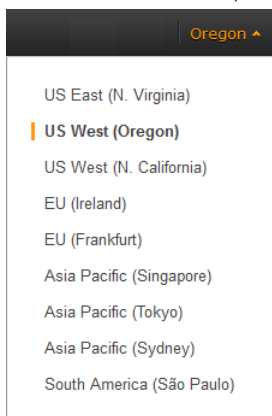
3. In the navigation pane, click **Alarms**.
4. Click **Create Alarm**, and then in the **CloudWatch Metrics by Category** pane, select a metric category, for example, **EBS Metrics**.
5. In the list of metrics, select a row that contains **VolumeWriteBytes** for a specific VolumeId.

   A graph showing average `VolumeWriteBytes` for a single volume appears in the lower pane.
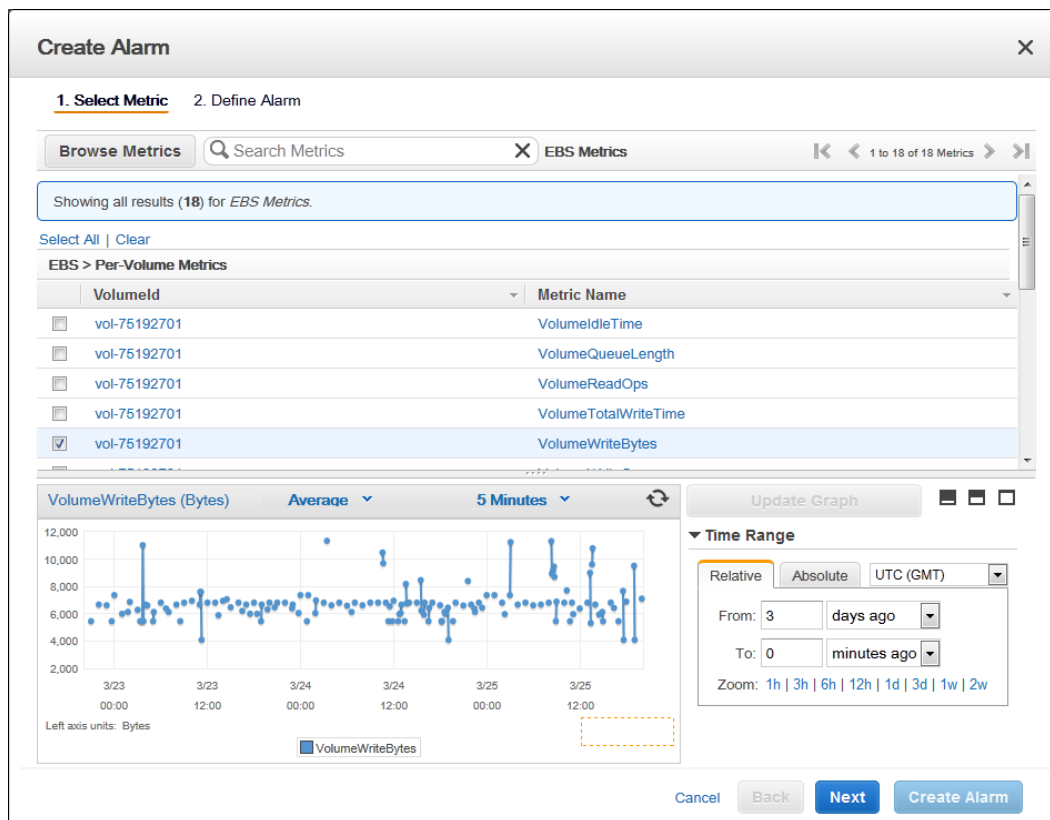
6. Select **Average** from the **Statistic** drop-down list.

7. Select **5 Minutes** from the **Period** drop-down list.

8. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: `myHighWriteAlarm`.

9. In the **Description** field, enter a description of the alarm, for example: `VolumeWriteBytes exceeds 100,000 KiB/s.`

10. In the **is** drop-down list, select **>**.

11. In the box next to the **is** drop-down list, enter `100000` and in the **for** field, enter `15`.

    A graphical representation of the threshold is shown under **Alarm Preview**.

12. Under **Actions**, in the **Whenever this alarm** drop-down list, select **State is ALARM**.

13. In the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.

14. To create a new Amazon SNS topic, select **New list**.

    In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: `myHigh-CpuAlarm`, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the `ALARM` state.

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

Command Line Interface

To send an Amazon Simple Notification Service email message when EBS exceeds 100 MB throughput

1. Create an Amazon Simple Notification Service topic. See instructions for creating an Amazon SNS topic in Set Up Amazon Simple Notification Service.

2. Use the `put-metric-alarm` command to create an alarm. For more information about the `put-metric-alarm` command, see put-metric-alarm in the *AWS Command Line Interface Reference*.

    ```
    $ aws cloudwatch put-metric-alarm --alarm-name ebs-mon --alarm-description
     "Alarm when EBS volume exceeds 100MB throughput" --metric-name VolumeRead
    Bytes --namespace AWS/EBS --statistic Average --period 300 --threshold
    100000000 --comparison-operator GreaterThanThreshold --dimensions
    Name=VolumeId,Value=my-volume-id --evaluation-periods 3 --alarm-actions
    arn:aws:sns:us-east-1:1234567890:my-alarm-topic --insufficient-data-actions
     arn:aws:sns:us-east-1:1234567890:my-insufficient-data-topic
    ```

    The AWS CLI returns to the command prompt if the command succeeds.

3. Test the alarm.

    *   Force an alarm state change to ALARM.

        ```
        $ aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason "ini
        tializing" --state-value OK
        $ aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason "ini
        tializing" --state-value ALARM
        $ aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason "ini
        tializing" --state-value INSUFFICIENT_DATA
        ```

    *   Check that two emails have been received.

Create Alarms That Stop or Terminate an Instance

Using Amazon CloudWatch alarm actions, you can create alarms that automatically stop or terminate your Amazon Elastic Compute Cloud (Amazon EC2) instances when you no longer need them to be running. For example, you might have instances dedicated to batch payroll processing jobs or scientific computing tasks that run for a period of time and then complete their work. Rather than leave those instances sitting idle (and accruing charges), you can stop or terminate them which can help you to save money. The main difference between using the stop and the terminate alarm actions is that you can easily restart a stopped instance if you need to run it again later, and you can keep the same instance ID and root volume. However, you cannot restart a terminated instance. Instead, you must launch a new instance.

You can add the stop or terminate alarm actions to any alarm that is set on an Amazon EC2 instance metric, including basic and detailed monitoring metrics provided by Amazon CloudWatch (in the AWS/EC2 namespace), as well as any custom metrics that include the "InstanceId=" dimension, as long as the InstanceId value refers to a valid running Amazon EC2 instance.

Contents

Adding Actions to Amazon CloudWatch Alarms

You can configure alarm actions using either the Amazon EC2 console or the Amazon CloudWatch console, or you can use the Amazon CloudWatch command line interface (CLI), API, or the AWS SDKs. For information about using the Amazon CloudWatch API with the AWS SDKs, see Sample Code & Libraries.

Using the Amazon EC2 Console to Create an Alarm to Stop an Instance

You can create an alarm that stops an Amazon EC2 instance when a certain threshold has been met. For example, you may run development or test instances and occasionally forget to shut them off. You can create an alarm that is triggered when the average CPU utilization percentage has been lower than 10 percent for 24 hours, signaling that it is idle and no longer in use. You can adjust the threshold, duration, and period to suit your needs, plus you can add an Amazon Simple Notification Service (Amazon SNS) notification, so that you will receive an email when the alarm is triggered.

Amazon EC2 instances that use an Amazon Elastic Block Store volume as the root device can be stopped or terminated, whereas instances that use the instance store as the root device can only be terminated.

> **Note**
> If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions: `ec2:DescribeInstanceStatus`, `ec2:DescribeInstances`, `ec2:StopInstances`, and `ec2:TerminateInstances` in order for the alarm action to be performed. If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see Permissions and Policies in *Using IAM*.
> If you are using an IAM role (e.g. Amazon EC2 instance profile), you cannot stop or terminate the instance using alarm actions. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.
> If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot stop or terminate an Amazon EC2 instance using alarm actions.

To create an alarm to stop an idle instance

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

2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see Regions and Endpoints.

3. In the navigation pane, under **INSTANCES**, click **Instances**.

4. In the upper pane, right-click an instance, and then click **Add/Edit Alarms**.

   Or, you can also select the instance, and then in the lower pane on the **Monitoring** tab, click **Create Alarm**.

5. In the **Alarm Details for** dialog box, click **Create Alarm**.

6. If you want to receive an email when the alarm is triggered, in the **Create Alarm for** dialog box, in the **Send a notification to** box, select an existing Amazon SNS topic, or click **Create Topic** to create a new one.

   If you create a new topic, in the **Send a notification to** box type a name for the topic, and then in the **With these recipients** box, type the email addresses of the recipients (separated by commas). Later, after you create the alarm, you will receive a subscription confirmation email that you must accept before you will get email for this topic.

7. Select the **Take the action** check box, and then choose the **Stop** radio button.

8. In the **Whenever** boxes, choose the statistic you want to use and then select the metric. In this example, choose **Average** and **CPU Utilization**.

9. In the **Is** boxes, define the metric threshold. In this example, enter **10** percent.

10. In the **For at least** box, choose the sampling period for the alarm. In this example, enter **24** consecutive periods of one hour.

11. To change the name of the alarm, in the **Name this alarm** box, type a new name.

    If you don't type a name for the alarm, Amazon CloudWatch automatically creates one for you.

    > **Note**
    > You can adjust the alarm configuration based on your own requirements before creating the alarm, or you can edit them later. This includes the metric, threshold, duration, action, and notification settings. However, after you create an alarm, you cannot edit its name later.

12. Click **Create Alarm**.

Using the Amazon EC2 Console to Create an Alarm that Terminates an Instance

You can create an alarm that terminates an EC2 instance automatically when a certain threshold has been met (as long as termination protection is not enabled for the instance). For example, you might want to terminate an instance when it has completed its work, and you don't need the instance again. If you might want to use the instance later, you should stop the instance instead of terminating it. For information about enabling and disabling termination protection for an instance, see Enabling Termination Protection for an Instance (p. 316).

> **Note**
> If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions: `ec2:DescribeInstanceStatus`, `ec2:DescribeInstances`, `ec2:StopInstances`, and `ec2:TerminateInstances` in order for the alarm action to be performed. If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see Permissions and Policies in *Using IAM*.
> If you are using an IAM role (e.g. Amazon EC2 instance profile), you cannot stop or terminate the instance using alarm actions. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.
> If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot stop or terminate an Amazon EC2 instance using alarm actions.

To create an alarm to terminate an idle instance

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

2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see Regions and Endpoints.

3. In the navigation pane, under **INSTANCES**, click **Instances**.

4. In the upper pane, right-click an instance, and then click **Add/Edit Alarms**.

    Or, select the instance and then in the lower pane, on the **Monitoring** tab, click **Create Alarm**.

5. In the **Alarm Details for** dialog box, click **Create Alarm**.

6. If you want to receive an email when the alarm is triggered, in the **Create Alarm for** dialog box, in the **Send a notification to** box, select an existing SNS topic, or click **Create Topic** to create a new one.

    If you create a new topic, in the **Send a notification to** box type a name for the topic, and then in the **With these recipients** box, type the email addresses of the recipients (separated by commas). Later, after you create the alarm, you will receive a subscription confirmation email that you must accept before you will get email for this topic.

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

    If you don't type a name for the alarm, Amazon CloudWatch automatically creates one for you.

    > **Note**
    > You can adjust the alarm configuration based on your own requirements before creating the alarm, or you can edit them later. This includes the metric, threshold, duration, action, and notification settings. However, after you create an alarm, you cannot edit its name later.

12. Click **Create Alarm**.

Using the Amazon CloudWatch Console to Create an Alarm that Stops an Instance

You can create an alarm that stops an Amazon EC2 instance when a certain threshold has been met. For example, you may run development or test instances and occasionally forget to shut them off. You can create an alarm that is triggered when the average CPU utilization percentage has been lower than 10 percent for 24 hours, signaling that it is idle and no longer in use. You can adjust the threshold, duration, and period to suit your needs, plus you can add an Amazon Simple Notification Service (Amazon SNS) notification, so that you will receive an email when the alarm is triggered.

Amazon CloudWatch alarm actions can stop an EBS-backed Amazon EC2 instances but they cannot stop instance store-backed Amazon EC2 instances. However, Amazon CloudWatch alarm actions can terminate either type of Amazon EC2 instance.

> **Note**
> If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions: `ec2:DescribeIn-stanceStatus`, `ec2:DescribeInstances`, `ec2:StopInstances`, and `ec2:TerminateIn-stances` in order for the alarm action to be performed. If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see Permissions and Policies in *Using IAM*.
> If you are using an IAM role (e.g. Amazon EC2 instance profile), you cannot stop or terminate the instance using alarm actions. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.
> If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot stop or terminate an Amazon EC2 instance using alarm actions.

To create an alarm to stop an idle instance

1. Open the Amazon CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see Regions and Endpoints.
3. In the navigation pane, click **Alarms**.
4. Click **Create Alarm**, and then in the **CloudWatch Metrics by Category** pane, under **EC2 Metrics**, select **Per-Instance Metrics**.
5. In the list of metrics, select the instance and metric you want to create an alarm for. You can also type an instance ID in the search box to go the instance that you want.
6. Select **Average** from the **Statistic** drop-down list.
7. Select a period from the **Period** drop-down list, for example: `1 Day`.
8. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: `stop EC2 instance`.
9. In the **Description** field, enter a description of the alarm, for example: `Stop EC2 instance when CPU is idle for too long.`
10. In the **is** drop-down list, select `<`.
11. In the box next to the **is** drop-down list, enter `10` and in the **for** field, enter `1440`.

    A graphical representation of the threshold is shown under **Alarm Preview**.
12. Under **Actions**, click **EC2 Action**.
13. In the **Whenever this alarm** drop-down list, select **State is ALARM**.

14. In the **Take this action** drop-down list, select **Stop this instance**.

15. Click **Notification**, and then in the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.

16. To create a new Amazon SNS topic, select **New list**.

    In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: **Stop_EC2_Instance**, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the ALARM state.

    > **Important**
    > If you are creating a new topic or adding email addresses to an existing topic, each email address that you add will be sent a topic subscription confirmation email. You must confirm the subscription by clicking the included link before notifications will be sent to a new email address.

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

Using the Amazon CloudWatch Console to Create an Alarm to Terminate an Idle Instance

You can create an alarm that terminates an Amazon EC2 instance automatically when a certain threshold has been met, as long as termination protection is disabled on the instance. For example, you might want to terminate an instance when it has completed its work, and you don't need the instance again. If you might want to use the instance later, you should stop the instance instead of terminating it. For information about disabling termination protection on an instance, see Enabling Termination Protection for an Instance (p. 316).

> **Note**
> If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions: ec2:DescribeInstanceStatus, ec2:DescribeInstances, ec2:StopInstances, and ec2:TerminateInstances in order for the alarm action to be performed. If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see Permissions and Policies in *Using IAM*.
> If you are using an IAM role (e.g. Amazon EC2 instance profile), you cannot stop or terminate the instance using alarm actions. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.
> If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot stop or terminate an Amazon EC2 instance using alarm actions.

To create an alarm to terminate an idle instance

1. Open the Amazon CloudWatch console at https://console.aws.amazon.com/cloudwatch/.

2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see Regions and Endpoints.

3. In the navigation pane, click **Alarms**.

4. In the upper pane, click **Create Alarm**.

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

6. In the list of metrics, select the instance and metric you want to create an alarm for. You can also type an instance ID in the search box to go the instance that you want.

7. Select **Average** from the **Statistic** drop-down list.

8. Select a period from the **Period** drop-down list, for example: **1 Day**.

9. Click **Next**, and then under **Alarm Threshold**, in the **Name** field, enter a unique name for the alarm, for example: `Terminate EC2 instance`.

10. In the **Description** field, enter a description of the alarm, for example: `Terminate EC2 instance when CPU is idle for too long.`

11. In the **is** drop-down list, select **<**.

12. In the box next to the **is** drop-down list, enter `10` and in the **for** field, enter `1440`.

    A graphical representation of the threshold is shown under **Alarm Preview**.

13. Under **Actions**, click EC2 Action.

14. In the **Whenever this alarm** drop-down list, select **State is ALARM**.

15. In the **Take this action** drop-down list, select **Terminate this instance**.

16. Click **Notification**, and then in the **Send notification to** drop-down list, select an existing Amazon SNS topic or create a new one.

17. To create a new Amazon SNS topic, select **New list**.

    In the **Send notification to** field, enter a name for the new Amazon SNS topic for example: `Terminate_EC2_Instance`, and in the **Email list** field, enter a comma-separated list of email addresses to be notified when the alarm changes to the `ALARM` state.

    > **Important**
    > If you are creating a new topic or adding email addresses to an existing topic, each email address that you add will be sent a topic subscription confirmation email. You must confirm the subscription by clicking the included link before notifications will be sent to a new email address.

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

Using the Amazon CloudWatch Console to View the History of Triggered Alarms and Actions

You can view alarm and action history in the Amazon CloudWatch console. Amazon CloudWatch keeps the last two weeks' worth of alarm and action history.

To view the history of triggered alarms and actions

1. Open the Amazon CloudWatch console at https://console.aws.amazon.com/cloudwatch/.

2. If necessary, change the region. From the navigation bar, select the region where your instance is running. For more information, see Regions and Endpoints.

3. In the navigation pane, click **Alarms**.

4. In the upper pane, select the alarm with the history that you want to view.

5. In the lower pane, the **Details** tab shows the most recent state transition along with the time and metric values.

6. Click the **History** tab to view the most recent history entries.

Using the CLI or the API to Create an Alarm to Stop or Terminate an Instance

If you are using either the AWS CLI or the CloudWatch API, or if you are using the AWS SDKs with the API, you can create an Amazon CloudWatch alarm using an Amazon EC2 instance metric, and then add an action using the action's dedicated Amazon Resource Name (ARN). You can add the action to any alarm state, and you can specify the region for each action. The region must match the region to which you send the put-metric-alarm request.

Action	ARN (with region)
Stop	arn:aws:automate:us-east-1:ec2:stop
Terminate	arn:aws:automate:us-east-1:ec2:terminate

For information about using the Amazon CloudWatch API with the AWS SDKs, see Sample Code & Libraries.

Note
If you are using an AWS Identity and Access Management (IAM) account to create or modify an alarm, you must have the following Amazon EC2 permissions: `ec2:DescribeInstanceStatus`, `ec2:DescribeInstances`, `ec2:StopInstances`, and `ec2:TerminateInstances` in order for the alarm action to be performed. If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see Permissions and Policies in *Using IAM*.
If you are using an IAM role (e.g. Amazon EC2 instance profile), you cannot stop or terminate the instance using alarm actions. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.
If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot stop or terminate an Amazon EC2 instance using alarm actions.

To create an alarm to stop an instance using the CLI

You can use the `arn:aws:automate:us-east-1:ec2:stop` ARN to stop an Amazon EC2 instance. The following example shows how to stop an instance if the average CPUUtilization is less than 10 percent over a 24 hour period.

*   At a command prompt, type:

```
% aws cloudwatch put-metric-alarm --alarm-name my-Alarm --alarm-description
 "Stop the instance when it is idle for a day" --namespace "AWS/EC2" --di
mensions Name=InstanceId,Value=i-abc123 --statistic Average --metric-name
CPUUtilization --comparison-operator LessThanThreshold --threshold 10 --
period 86400 --evaluation-periods 4 --alarm-actions arn:aws:automate:us-
east-1:ec2:stop
```

To create an alarm to terminate an instance using the CLI

*   At a command prompt, type:

```
% aws cloudwatch put-metric-alarm --alarm-name my-Alarm --alarm-description
 "Terminate the instance when it is idle for a day" --namespace "AWS/EC2"
--dimensions Name=InstanceId,Value=i-abc123" --statistic Average  --metric-
name CPUUtilization --comparison-operator LessThanThreshold --threshold 1
--period 86400 --evaluation-periods 4 -- alarm-actions arn:aws:automate:us-
east-1:ec2:terminate
```

To create an alarm and to stop an instance using the API

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

• Construct the following request:

```
http://monitoring.amazonaws.com/

?SignatureVersion=2

&Action=PutMetricAlarm

&Version=2009-05-15

&Namespace=AWS/EC2

&MetricName=CPUUtilization

&Dimension.member.1.Name=instance-id

&Dimension.member.1.Value=i-abc123

&Period=86400

&Statistic=Average

&AlarmName=Stop-EC2-Instance

&ComparisonOperator=LessThanThreshold

&Threshold=10

&EvaluationPeriods=4

&StartTime=2009-01-16T00:00:00

&EndTime=2009-01-16T00:02:00

&Timestamp=2009-01-08-18

&AWSAccessKeyId=XXX YOUR ACCESS KEY XXX

&Signature=%XXX YOUR SIGNATURE XXX%3D

&AlarmActions.member.1=arn:aws:automate:us-east-1:ec2:stop
```

To create an alarm and to terminate an instance using the API

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

• Construct the following request:

```
http://monitoring.amazonaws.com/

?SignatureVersion=2

```

```
&Action=PutMetricAlarm

&Version=2009-05-15

&Namespace=AWS/EC2

&MetricName=CPUUtilization

&Dimension.member.1.Name=instance-id

&Dimension.member.1.Value=i-abc123

&Period=86400

&Statistic=Average

&AlarmName=Terminate-EC2-Instance

&ComparisonOperator=LessThanThreshold

&Threshold=10

&EvaluationPeriods=4

&StartTime=2009-01-16T00:00:00

&EndTime=2009-01-16T00:02:00

&Timestamp=2009-01-08-18

&AWSAccessKeyId=XXX YOUR ACCESS KEY XXX

&Signature=%XXX YOUR SIGNATURE XXX%3D

&AlarmActions.member.1=arn:aws:automate:us-east-1:ec2:terminate
```

Amazon CloudWatch Alarm Action Scenarios

You can use the Amazon Elastic Compute Cloud (Amazon EC2) console to create alarm actions that
stop or terminate an Amazon EC2 instance when certain conditions are met. In the following screen
capture of the console page where you set the alarm actions, we've numbered the settings. We've also
numbered the settings in the scenarios that follow, to help you create the appropriate actions.

Scenario 1: Stop Idle Development and Test Instances

Create an alarm that stops an instance used for software development or testing when it has been idle for at least an hour.

Setting	Value
☒	Stop
☒	Maximum
☒	CPUUtilization
☒	<=
☒	10%
☒	60 minutes
☒	1

Scenario 2: Stop Idle Instances

Create an alarm that stops an instance and sends an email when the instance has been idle for 24 hours.

Setting	Value
☒	Stop and email
☒	Average
☒	CPUUtilization

Setting	Value
![x]	<=
![x]	5%
![x]	60 minutes
![x]	24

Scenario 3: Stop Web Servers with Unusually High Traffic

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

Setting	Value
![x]	Email
![x]	Sum
![x]	NetworkOut
![x]	>
![x]	10 GB
![x]	1 day
![x]	1

Scenario 4: Stop Web Servers with Unusually High Traffic

Create an alarm that stops an instance and send a text message (SMS) if outbound traffic exceeds 1 GB per hour.

Setting	Value
![x]	Stop and send SMS
![x]	Sum
![x]	NetworkOut

Setting	Value
	>
	1 GB
	1 hour
	1

Scenario 5: Stop an Instance Experiencing a Memory Leak

Create an alarm that stops an instance when memory utilization reaches or exceeds 90 percent, so that application logs can be retrieved for troubleshooting.

> **Note**
> The MemoryUtilization metric is a custom metric. In order to use the MemoryUtilization metric, you must install the Monitoring Scripts for Amazon EC2 Instances (p. 405).

Setting	Value
	Stop
	Maximum
	MemoryUtilization
	>=
	90%
	1 minute
	1

Scenario 6: Stop an Impaired Instance

Create an alarm that stops an instance that fails three consecutive status checks (performed at 5-minute intervals).

Setting	Value
	Stop

Setting	Value
![x]	Average
![x]	StatusCheckFailed_System
![x]	>=
![x]	1
![x]	15 minutes
![x]	1

Scenario 7: Terminate Instances When Batch Processing Jobs Are Complete

Create an alarm that terminates an instance that runs batch jobs when it is no longer sending results data.

Setting	Value
![x]	Terminate
![x]	Maximum
![x]	NetworkOut
![x]	<=
![x]	100,000 bytes
![x]	5 minutes
![x]	1

The previous scenarios can also be performed using the Amazon CloudWatch console. We've numbered the settings on the console to match the numbered settings in the Amazon EC2 console and the scenarios that we covered earlier, so you can make a comparison and create an alarm with the appropriate actions.

Monitoring Scripts for Amazon EC2 Instances

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

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

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

Amazon CloudWatch Monitoring Scripts for Linux

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

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

- Amazon Linux 2011.09.02

- Red Hat Enterprise Linux 6.3
- Ubuntu Server 12.04
- SUSE Linux Enterprise Server 11

Contents

Prerequisites

You must perform additional steps on some versions of Linux.

Amazon Linux AMI

If you are running an Amazon Linux AMI, version 2014.03 or later, you'll need to add some additional
Perl modules in order for the monitoring scripts to work. Use the following procedure to configure your
server.

- Log on to your Amazon Linux AMI instance and install the following packages:

```
$ sudo yum install perl-Switch perl-Sys-Syslog perl-LWP-Protocol-https
```

Red Hat Enterprise Linux or SUSE Linux

If you are running Red Hat Enterprise Linux or SUSE Linux Enterprise Server, use the following procedure
to configure your server.

1. At a command prompt, type `cd /` to switch to the root directory.
2. Type `perl -MCPAN -e shell`, and then type `yes` at every prompt.
3. At the `cpan[1]>` prompt, type `install Bundle::LWP5_837 LWP` to install the LWP bundle and
    update to LWP version 6.03. Type `yes` at every prompt.

Ubuntu Server

If you are running Ubuntu Server, use the following procedure to configure your server.

- Log on to your Ubuntu Server instance and install the following packages:

```
sudo apt-get install unzip libwww-perl libcrypt-ssleay-perl
```

```
sudo apt-get install libswitch-perl
```

For information about connecting to Amazon EC2 Linux instances, see Connect to Your Linux In-
stance (p. 299).

Getting Started

The following steps show you how to download, uncompress, and configure the CloudWatch Monitoring Scripts on an EC2 Linux instance.

To download, install, and configure the script

- Open a command prompt, move to a folder where you want to store the scripts and enter the following:

```
wget http://ec2-downloads.s3.amazonaws.com/cloudwatch-samples/CloudWatchMon
itoringScripts-v1.1.0.zip
unzip CloudWatchMonitoringScripts-v1.1.0.zip
rm CloudWatchMonitoringScripts-v1.1.0.zip
cd aws-scripts-mon
```

The `CloudWatchMonitoringScripts-v1.1.0.zip` package contains these files:

- **CloudWatchClient.pm**—Shared Perl module that simplifies calling Amazon CloudWatch from other scripts.
- **mon-put-instance-data.pl**—Collects system metrics on an Amazon EC2 instance (memory, swap, disk space utilization) and sends them to Amazon CloudWatch.
- **mon-get-instance-stats.pl**—Queries Amazon CloudWatch and displays the most recent utilization statistics for the EC2 instance on which this script is executed.
- —File template for AWS credentials that stores your access key ID and secret access key.
- **LICENSE.txt**—Text file containing the Apache 2.0 license.
- **NOTICE.txt**—copyright notice.

mon-put-instance-data.pl

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

Options

Name	Description
`--mem-util`	Collects and sends the MemoryUtilization metrics in percentages. This option reports only memory allocated by applications and the operating system, and excludes memory in cache and buffers.
`--mem-used`	Collects and sends the MemoryUsed metrics, reported in megabytes. This option reports only memory allocated by applications and the operating system, and excludes memory in cache and buffers.
`--mem-avail`	Collects and sends the MemoryAvailable metrics, reported in megabytes. This option reports memory available for use by applications and the operating system.
`--swap-util`	Collects and sends SwapUtilization metrics, reported in percentages.
`--swap-used`	Collects and sends SwapUsed metrics, reported in megabytes.

Name	Description
`--disk-path=PATH`	Selects the disk on which to report. PATH can specify a mount point or any file located on a mount point for the filesystem that needs to be reported. For selecting multiple disks, specify a `--disk-path=PATH` for each one of them. To select a disk for the filesystems mounted on `/` and `/home`, use the following parameters: `--disk-path=/ --disk-path=/home`
`--disk-space-util`	Collects and sends the DiskSpaceUtilization metric for the selected disks. The metric is reported in percentages.
`--disk-space-used`	Collects and sends the DiskSpaceUsed metric for the selected disks. The metric is reported by default in gigabytes. Due to reserved disk space in Linux operating systems, disk space used and disk space available may not accurately add up to the amount of total disk space.
`--disk-space-avail`	Collects and sends the DiskSpaceAvailable metric for the selected disks. The metric is reported in gigabytes. Due to reserved disk space in the Linux operating systems, disk space used and disk space available may not accurately add up to the amount of total disk space.
`--memory-units=UNITS`	Specifies units in which to report memory usage. If not specified, memory is reported in megabytes. UNITS may be one of the following: bytes, kilobytes, megabytes, gigabytes.
`--disk-space-units=UNITS`	Specifies units in which to report disk space usage. If not specified, disk space is reported in gigabytes. UNITS may be one of the following: bytes, kilobytes, megabytes, gigabytes.
`--aws-credential-file=PATH`	Provides the location of the file containing AWS credentials. This parameter cannot be used with the `--aws-access-key-id` and `--aws-secret-key` parameters.
`--aws-access-key-id=VALUE`	Specifies the AWS access key ID to use to identify the caller. Must be used together with the `--aws-secret-key` option. Do not use this option with the `--aws-credential-file` parameter.
`--aws-secret-key=VALUE`	Specifies the AWS secret access key to use to sign the request to CloudWatch. Must be used together with the `--aws-access-key-id` option. Do not use this option with `--aws-credential-file` parameter.
`--verify`	Performs a test run of the script that collects the metrics, prepares a complete HTTP request, but does not actually call CloudWatch to report the data. This option also checks that credentials are provided. When run in verbose mode, this option outputs the metrics that will be sent to CloudWatch.
`--from-cron`	Use this option when calling the script from cron. When this option is used, all diagnostic output is suppressed, but error messages are sent to the local system log of the user account.

Name	Description
`--verbose`	Displays detailed information about what the script is doing.
`--help`	Displays usage information.
`--version`	Displays the version number of the script.

Examples

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

To perform a simple test run without posting data to CloudWatch

• Run the following command:

```
./mon-put-instance-data.pl --mem-util --verify --verbose
```

To collect all available memory metrics and send them to CloudWatch

• Run the following command:

```
./mon-put-instance-data.pl --mem-util --mem-used --mem-avail
```

To set a cron schedule for metrics reported to CloudWatch

1. Start editing the crontab using the following command:

```
crontab -e
```

2. Add the following command to report memory and disk space utilization to CloudWatch every five minutes:

```
*/5 * * * * ~/aws-scripts-mon/mon-put-instance-data.pl --mem-util --disk-space-util --disk-path=/ --from-cron
```

If the script encounters an error, the script will write the error message in the system log.

mon-get-instance-stats.pl

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

Options

Name	Description
`--recent-hours=N`	Specifies the number of recent hours to report on, as represented by `N` where N is an integer.
`--aws-credential-file=PATH`	Provides the location of the file containing AWS credentials.
`--aws-access-key-id=VALUE`	Specifies the AWS access key ID to use to identify the caller. Must be used together with the `--aws-secret-key` option. Do not use this option with the `--aws-credential-file` option.
`--aws-secret-key=VALUE`	Specifies the AWS secret access key to use to sign the request to CloudWatch. Must be used together with the `--aws-access-key-id` option. Do not use this option with `--aws-credential-file` option.
`--verify`	Performs a test run of the script that collects the metrics, prepares a complete HTTP request, but does not actually call CloudWatch to report the data. This option also checks that credentials are provided. When run in verbose mode, this option outputs the metrics that will be sent to CloudWatch.
`--verbose`	Displays detailed information about what the script is doing.
`--help`	Displays usage information.
`--version`	Displays the version number of the script.

Examples

To get utilization statistics for the last 12 hours

- Run the following command:

```
mon-get-instance-stats.pl --recent-hours=12
```

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

```
Instance metric statistics for the last 12 hours.

CPU Utilization
    Average: 1.06%, Minimum: 0.00%, Maximum: 15.22%

Memory Utilization
    Average: 6.84%, Minimum: 6.82%, Maximum: 6.89%

Swap Utilization
    Average: N/A, Minimum: N/A, Maximum: N/A

Disk Space Utilization on /dev/xvda1 mounted as /
    Average: 9.69%, Minimum: 9.69%, Maximum: 9.69%
```

Viewing Your Custom Metrics in the Console

If you successfully call the mon-put-instance-data.pl script, you can use the AWS Management Console to view your posted custom metrics in the Amazon CloudWatch console.

To view custom metrics

1. Execute mon-put-instance-data.pl, as described earlier.
2. Sign in to the AWS Management Console and open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
3. Click **View Metrics**.
4. In the **Viewing** drop-down list, your custom metrics posted by the script display with the prefix System/Linux.

Network and Security

Amazon EC2 provides the following network and security features.

Features

- Amazon EC2 Key Pairs (p. 413)
- Amazon EC2 Security Groups (p. 421)
- Controlling Access to Amazon EC2 Resources (p. 429)
- Amazon EC2 and Amazon Virtual Private Cloud (VPC) (p. 468)
- Amazon EC2 Instance IP Addressing (p. 480)
- Elastic IP Addresses (EIP) (p. 489)
- Elastic Network Interfaces (ENI) (p. 494)
- Enabling Enhanced Networking on Linux Instances in a VPC (p. 507)

If you access Amazon EC2 using the command line tools or an API, you'll need your access key ID and secret access key. For more information, see How Do I Get Security Credentials? in the *Amazon Web Services General Reference*.

You can launch an instance into one of two platforms: EC2-Classic or EC2-VPC. An instance that's launched into EC2-Classic or a default VPC is automatically assigned a public IP address. An instance that's launched into a nondefault VPC can be assigned a public IP address on launch. For more information about EC2-Classic and EC2-VPC, see Supported Platforms (p. 471).

Instances can fail or terminate for reasons outside of your control. If an instance fails and you launch a replacement instance, the replacement has a different public IP address than the original. However, if your application needs a static IP address, you can use an *Elastic IP address*.

You can use *security groups* to control who can access your instances. These are analogous to an inbound network firewall that enables you to specify the protocols, ports, and source IP ranges that are allowed to reach your instances. You can create multiple security groups and assign different rules to each group. You can then assign each instance to one or more security groups, and we use the rules to determine which traffic is allowed to reach the instance. You can configure a security group so that only specific IP addresses or specific security groups have access to the instance.

Amazon EC2 Key Pairs

Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. Public–key cryptography uses a public key to encrypt a piece of data, such as a password, then the recipient uses the private key to decrypt the data. The public and private keys are known as a *key pair*.

To log in to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance. Linux instances have no password, and you use a key pair to log in using SSH.

Creating a Key Pair

You can use Amazon EC2 to create your key pair. For more information, see Creating Your Key Pair Using Amazon EC2 (p. 413).

Alternatively, you could use a third-party tool and then import the public key to Amazon EC2. For more information, see Importing Your Own Key Pair to Amazon EC2 (p. 415).

Each key pair requires a name. Be sure to choose a name that is easy to remember. Amazon EC2 associates the public key with the name that you specify as the key name.

Amazon EC2 stores the public key only, and you store the private key. Anyone who possesses your private key can decrypt your login information, so it's important that you store your private keys in a secure place.

The keys that Amazon EC2 uses are 1024-bit SSH-2 RSA keys. You can have up to five thousand key pairs per region.

Launching and Connecting to Your Instance

When you launch an instance, you should specify the name of the key pair you plan to use to connect to the instance. If you don't specify the name of an existing key pair when you launch an instance, you won't be able to connect to the instance. When you connect to the instance, you must specify the private key that corresponds to the key pair you specified when you launched the instance. Amazon EC2 doesn't keep a copy of your private key; therefore, if you lose your private key, there is no way to recover it. If you lose the private key for an instance store-backed instance, you can't access the instance; you should terminate the instance and launch another instance using a new key pair. If you lose the private key for an EBS-backed instance, you can regain access to your instance. For more information, see Connecting to Your Linux Instance if You Lose Your Private Key (p. 418).

Topics

Creating Your Key Pair Using Amazon EC2

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

To create your key pair using the console

1. Open the Amazon EC2 console.

2. From the navigation bar, select a region for the key pair. You can select any region that's available to you, regardless of your location. This choice is important because some Amazon EC2 resources can be shared between regions, but key pairs can't. For example, if you create a key pair in the US West (Oregon) region, you can't see or use the key pair in another region.

3. Click **Key Pairs** in the navigation pane.

4. Click **Create Key Pair**.

5. Enter a name for the new key pair in the **Key pair name** field of the **Create Key Pair** dialog box, and then click **Create**.

6. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is `.pem`. Save the private key file in a safe place.

   Important
   This is the only chance for you to save the private key file. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

7. If you will use an SSH client on a Mac or Linux computer to connect to your Linux instance, use the following command to set the permissions of your private key file so that only you can read it.

```
$ chmod 400 my-key-pair.pem
```

To create your key pair using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- create-key-pair (AWS CLI)
- ec2-create-keypair (Amazon EC2 CLI)
- New-EC2KeyPair (AWS Tools for Windows PowerShell)

Importing Your Own Key Pair to Amazon EC2

If you used Amazon EC2 to create your key pair, as described in the previous section, you are ready to launch an instance. Otherwise, instead of using Amazon EC2 to create your key pair, you can create an RSA key pair using a third-party tool and then import the public key to Amazon EC2. For example, you can use **ssh-keygen** (a tool provided with the standard OpenSSH installation) to create a key pair. Alternatively, Java, Ruby, Python, and many other programming languages provide standard libraries that you can use to create an RSA key pair.

Amazon EC2 accepts the following formats:

* OpenSSH public key format (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.

To create a key pair using a third-party tool

1. Generate a key pair with a third-party tool of your choice.
2. Save the public key to a local file. For example, `~/.ssh/my-key-pair.pub`. The file name extension for this file is not important.
3. Save the private key to a different local file that has the `.pem` extension. For example, `~/.ssh/my-key-pair.pem`. Save the private key file in a safe place. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

Use the following steps to import your key pair using the Amazon EC2 console. (If you prefer, you can use the ec2-import-keypair command or the ImportKeyPair action to import the public key.)

To import the public key

1. Open the Amazon EC2 console.
2. From the navigation bar, select the region for the key pair. This choice is important because key pair resources cannot be shared between regions. For example, if you import a key pair into the US West (Oregon) region, you won't be able to see or use the key pair in another region.
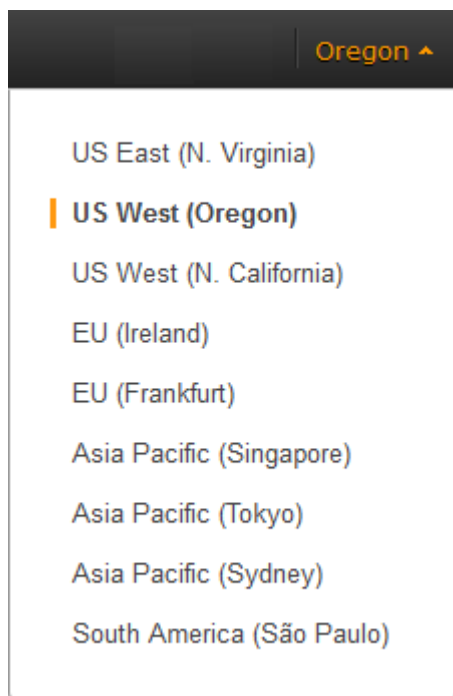
3. Click **Key Pairs** in the navigation pane.
4. Click **Import Key Pair**.
5. In the **Import Key Pair** dialog box, click **Browse**, and select the public key file that you saved previously. Enter a name for the key pair in the **Key pair name** field, and click **Import**.

After the public key file is imported, you can verify that the key pair was imported successfully using the Amazon EC2 console as follows. (If you prefer, you can use the ec2-describe-keypairs command or the DescribeKeyPairs action to list your key pairs.)

To verify that your key pair was imported

1. From the navigation bar, select the region in which you created the key pair.
2. Click **Key Pairs** in the navigation pane.
3. Verify that the key pair that you imported is in the displayed list of key pairs.

Retrieving the Public Key for Your Key Pair

On a Linux instance, the public key content is placed in an entry within `~/.ssh/authorized_keys`. This is done at boot time and enables you to securely access your instance without passwords. You can open this file in an editor to view the public key for your key pair. The following is an example entry for the key pair named my-key-pair. It consists of the public key followed by the name of the key pair. For example:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

You can use **ssh-keygen** to get the public key for your key pair. Run the following command on a computer to which you've downloaded your private key:

```
$ ssh-keygen -y
```

When prompted to enter the file in which the key is, specify the path to your `.pem` file; for example:

```
/path_to_key_pair/my-key-pair.pem
```

The command returns the public key:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

If this command fails, ensure that you've changed the permissions on your key pair file so that only you can view it by running the following command:

```
$ chmod 400 my-key-pair.pem
```

The public key that you specified when you launched an instance is also available to you through its instance metadata. To view the public key that you specified when launching the instance, use the following command from your instance:

```
$ GET http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

For more information, see Retrieving Instance Metadata (p. 257).

Note that if you change the key pair that you use to connect to the instance, as shown in the last section on this page, we don't update the instance metadata to show the new public key; you'll continue to see the public key for the key pair you specified when you launched the instance in the instance metadata.

Verifying Your Key Pair's Fingerprint

On the **Key Pairs** page in the Amazon EC2 console, the **Fingerprint** column displays the fingerprints generated from your key pairs. AWS calculates the fingerprint differently depending on whether the key pair was generated by AWS or a third-party tool. If you created the key pair using AWS, the fingerprint is calculated using an SHA-1 hash function. If you created the key pair with a third-party tool and uploaded the public key to AWS, or if you generated a new public key from an existing AWS-created private key and uploaded it to AWS, the fingerprint is calculated using an MD5 hash function.

You can use the fingerprint that's displayed on the **Key Pairs** page to verify that the private key you have on your local machine matches the public key that's stored in AWS.

If you created your key pair using AWS, you can use the ec2-fingerprint-key command in the Amazon EC2 CLI to generate a fingerprint from the private key file on your local machine. The output should match

the fingerprint that's displayed in the console. Alternatively, you can use the OpenSSL tools to generate a fingerprint from the private key file:

```
$ openssl pkcs8 -in path_to_private_key -inform PEM -outform DER -topk8 -nocrypt
 | openssl sha1 -c
```

If you created your key pair using a third-party tool and uploaded the public key to AWS, you can use the OpenSSL tools to generate a fingerprint from the private key file on your local machine:

```
$ openssl rsa -in path_to_private_key -pubout -outform DER | openssl md5 -c
```

The output should match the fingerprint that's displayed in the console.

Deleting Your Key Pair

When you delete a key pair, you are only deleting Amazon EC2's copy of the public key. Deleting a key pair doesn't affect the private key on your computer or the public key on any instances already launched using that key pair. You can't launch a new instance using a deleted key pair, but you can continue to connect to any instances that you launched using a deleted key pair, as long as you still have the private key (`.pem`) file.

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

To delete your key pair using the console

1. Open the Amazon EC2 console.
2. Click **Key Pairs** in the navigation pane.
3. Select the key pair and click **Delete**.
4. When prompted, click **Yes**.

To delete your key pair using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- delete-key-pair (AWS CLI)
- ec2-delete-keypair (Amazon EC2 CLI)
- Remove-EC2KeyPair (AWS Tools for Windows PowerShell)

Connecting to Your Linux Instance if You Lose Your Private Key

If you lose the private key for an EBS-backed instance, you can regain access to your instance. You must stop the instance, detach its root volume and attach it to another instance as a data volume, modify the `authorized_keys` file, move the volume back to the original instance, and restart the instance. For more information about launching, connecting to, and stopping instances, see Instance Lifecycle (p. 286).

This procedure isn't supported for instance store-backed instances. To determine the root device type of your instance, open the Amazon EC2 console, click **Instances**, select the instance, and check the value of **Root device type** in the details pane. The value is either `ebs` or `instance store`. If the root device is an instance store volume, you must have the private key in order to connect to the instance.

Tip

If the root volume of the instance has an AWS Marketplace product code, you can only attach it to an instance as the root device, not a secondary device. To determine whether a volume has an AWS Marketplace product code, select the instance, and check the value of **Product codes** in the details pane. If your volume has an AWS Marketplace product code and you need to complete this procedure, create a snapshot of the EBS volume, and then post a request to the Amazon EC2 forum to recover a instance with an AWS Marketplace product code. Include the Availability Zone, instance ID, AMI ID, and snapshot ID in the request. Someone from support will respond with further directions.

Prerequisites

Create a new key pair using either the Amazon EC2 console or a third-party tool.

To connect to an EBS-backed instance with a different key pair

1. From the Amazon EC2 console, click **Instances** in the navigation pane, and select the instance that you'd like to connect to. (We'll refer to this as the original instance.)
2. Save the following information that you'll see to complete this procedure.

   - Write down the instance ID (i-*xxxxxxxx*), AMI ID (ami-*xxxxxxxx*), and Availability Zone of the original instance.
   - Click the entry for `sda1` (the root volume) under **Block devices** in the details pane and write down the volume ID (vol-*xxxxxxxx*).
   - [EC2-Classic] If the original instance has an associated Elastic IP address, write down the Elastic IP address shown under **Elastic IP** in the details pane.

3. Click **Actions**, and then click **Stop**. If **Stop** is disabled, either the instance is already stopped or its root device is an instance store volume.
4. Launch a temporary instance using the console with the following options:

   - On the **Choose an AMI** page, select the same AMI that you used to launch the original instance. If this AMI is unavailable, you can create an AMI that you can use from the stopped instance. For more information, see Creating an Amazon EBS-Backed Linux AMI (p. 74) .
   - On the **Choose an Instance Type** page, leave the default instance type that the wizard selects for you.
   - On the **Configure Instance Details** page, specify the same Availability Zone as the instance you'd like to connect to. If you're launching an instance in a VPC, select a subnet in this Availability Zone.
   - On the **Tag Instance** page, add the tag `Name=Temporary` to the instance to indicate that this is a temporary instance.
   - On the **Review** page, click **Launch**. Specify the new key pair that you created, and then click **Launch Instances**.

5. In the navigation pane, click **Volumes** and select the root device volume for the original instance (you wrote down its volume ID in a previous step). Click **Actions**, and then click **Detach Volume**. Wait for the state of the volume to become `available`. (You might need to click the **Refresh** icon.)
6. With the volume still selected, click **Actions**, and then click **Attach Volume**. Select the instance ID of the temporary instance, write down the device name specified under **Device** (for example, `/dev/sdf`), and then click **Yes, Attach**.

   > **Note**
   >
   > If the volume has an AWS Marketplace product code, you'll get an error indicating that you can't attach the volume as a secondary device. For more information, see the tip at the start of this section.

7. Connect to the temporary instance.

8. From the temporary instance, mount the volume that you attached to the instance so that you can access its file system. For example, if the device name is `/dev/sdf`, use the following commands to mount the volume as `/mnt/tempvol`.

   Note
   The device name may appear differently on your instance. For example, devices mounted as `/dev/sdf` may show up as `/dev/xvdf` on the instance. Some versions of Red Hat (or its variants, such as CentOS) may even increment the trailing letter by 4 characters, where `/dev/sdf` becomes `/dev/xvdk`.

   a. Use the **lsblk** command to determine if the volume is partitioned.

   ```
   [ec2-user ~]$ lsblk
   NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
   xvda    202:0    0    8G  0 disk
     xvda1 202:1    0    8G  0 part /
   xvdf    202:80   0  101G  0 disk
     xvdf1 202:81   0  101G  0 part
   xvdg    202:96   0   30G  0 disk
   ```

   In the above example, `/dev/xvda` and `/dev/xvdf` are partitioned volumes, and `/dev/xvdg` is not. If your volume is partitioned, you mount the partition (`/dev/xvdf1`) instead of the raw device (`/dev/xvdf`) in the next steps.

   b. Create a temporary directory to mount the volume.

   ```
   [ec2-user ~]$ sudo mkdir /mnt/tempvol
   ```

   c. Mount the volume (or partition) at the temporary mount point, using the volume name or device name you identified earlier.

   ```
   [ec2-user ~]$ sudo mount /dev/xvdf1 /mnt/tempvol
   ```

9. From the temporary instance, use the following command to update `authorized_keys` on the mounted volume with the new public key from the `authorized_keys` for the temporary instance (you may need to substitute a different user name in the following command, such as `ubuntu` for Ubuntu instances):

   ```
   [ec2-user ~]$ cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/author
   ized_keys
   ```

   If this copy succeeded, you can go to the next step.

   (Optional) Otherwise, if you don't have permission to edit files in `/mnt/tempvol`, you'll need to update the file using **sudo** and then check the permissions on the file to verify that you'll be able to log into the original instance. Use the following command to check the permissions on the file:

   ```
   [ec2-user ~]$ sudo ls -l /mnt/home/ec2-user/.ssh
   total 4
   -rw------- 1 222 500 398 Sep 13 22:54 authorized_keys
   ```

In this example output, *222* is the user ID and *500* is the group ID. Next, use **sudo** to re-run the copy command that failed:

```
[ec2-user ~]$ sudo cp .ssh/authorized_keys /mnt/tempvol/home/ec2-
user/.ssh/authorized_keys
```

Run the following command again to determine whether the permissions changed:

```
[ec2-user ~]$ sudo ls -l /mnt/home/ec2-user/.ssh
```

If the user ID and group ID have changed, use the following command to restore them:

```
[ec2-user ~]$ sudo chown 222:500 /mnt/home/ec2-user/.ssh/authorized_keys
```

10. From the temporary instance, unmount the volume that you attached so that you can reattach it to the original instance. For example, use the following command to unmount the volume at `/mnt/tempvol`:

```
[ec2-user ~]$ sudo umount /mnt/tempvol
```

11. From the Amazon EC2 console, select the volume with the volume ID that you wrote down, click **Actions**, and then click **Detach Volume**. Wait for the state of the volume to become `available`. (You might need to click the **Refresh** icon.)

12. With the volume still selected, click **Actions**, and then click **Attach Volume**. Select the instance ID of the original instance, specify the device name `/dev/sda1`, and then click **Yes, Attach**.

> **Warning**
> If you don't specify `sda1` as the device name, you'll be unable to start the original instance. This is because Amazon EC2 expects the root device volume at `sda1`.

13. Select the original instance, click **Actions**, and then click **Start**. After the instance enters the `running` state, you can connect to it using the private key file for your new key pair.

14. [EC2-Classic] If the original instance had an associated Elastic IP address before you stopped it, you must re-associate it with the instance as follows:

    a. In the navigation pane, click **Elastic IPs**.

    b. Select the Elastic IP address that you wrote down at the beginning of this procedure.

    c. Click **Associate Address**.

    d. Select the ID of the original instance, and then click **Associate**.

15. (Optional) You can terminate the temporary instance if you have no further use for it. Select the temporary instance, click **Actions**, and then click **Terminate**.

Amazon EC2 Security Groups

A *security group* acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group. When we decide whether to allow traffic to reach an instance, we evaluate all the rules from all the security groups that are associated with the instance.

Topics

If you have requirements that aren't met by security groups, you can maintain your own firewall on any of your instances in addition to using security groups.

Security Groups for EC2-Classic

If you're using EC2-Classic, you must use security groups created specifically for EC2-Classic. When you launch an instance in EC2-Classic, you must specify a security group in the same region as the instance. You can't specify a security group that you created for a VPC when you launch an instance in EC2-Classic.

After you launch an instance in EC2-Classic, you can't change its security groups. However, you can add rules to or remove rules from a security group, and those changes are automatically applied to all instances that are associated with the security group.

> **Note**
> In EC2-Classic, you can associate an instance with up to 500 security groups and add up to 100 rules to a security group.

Security Groups for EC2-VPC

If you're using EC2-VPC, you must use security groups created specifically for your VPC. When you launch an instance in a VPC, you must specify a security group for that VPC. You can't specify a security group that you created for EC2-Classic when you launch an instance in a VPC.

After you launch an instance in a VPC, you can change its security groups. You can also change the rules of a security group, and those changes are automatically applied to all instances that are associated with the security group.

> **Note**
> In EC2-VPC, you can associate a network interface with up to 5 security groups and add up to 50 rules to a security group.

When you specify a security group for a nondefault VPC to the CLI or the API actions, you must use the security group ID and not the security group name to identify the security group.

Security groups for EC2-VPC have additional capabilities that aren't supported by security groups for EC2-Classic. For more information about security groups for EC2-VPC, see Security Groups for Your VPC in the *Amazon VPC User Guide*.

Security Group Rules

The rules of a security group control the inbound traffic that's allowed to reach the instances that are associated with the security group and the outbound traffic that's allowed to leave them. By default, security groups allow all outbound traffic.

You can add and remove rules at any time. Your changes are automatically applied to the instances associated with the security group after a short period. You can either edit an existing rule in a security group, or delete it and add a new rule. You can copy the rules from an existing security group to a new security group. You can't change the outbound rules for EC2-Classic. Security group rules are always permissive; you can't create rules that deny access.

For each rule, you specify the following:

- The protocol to allow (such as TCP, UDP, or ICMP).
- TCP and UDP, or a custom protocol: The range of ports to allow
- ICMP: The ICMP type and code
- One or the following options for the source (inbound rules) or destination (outbound rules):
  - An individual IP address, in CIDR notation. Be sure to use the `/32` prefix after the IP address; if you use the `/0` prefix after the IP address, this opens the port to everyone. For example, specify the IP address `203.0.113.1` as `203.0.113.1/32`.
  - An IP address range, in CIDR notation (for example, `203.0.113.0/24`).
  - The name (EC2-Classic) or ID (EC2-Classic or EC2-VPC) of a security group. This allows instances associated with the specified security group to access instances associated with this security group. (Note that this does not add rules from the source security group to this security group.) You can specify one of the following security groups:
    - The current security group.
    - EC2-Classic: A different security group for EC2-Classic in the same region
    - EC2-VPC: A different security group for the same VPC
    - EC2-Classic: A security group for another AWS account in the same region (add the AWS account ID as a prefix; for example, `111122223333/sg-edcd9784`)

When you specify a security group as the source or destination for a rule, the rule affects all instances associated with the security group. Incoming traffic is allowed based on the private IP addresses of the instances that are associated with the source security group (and not the public IP or Elastic IP addresses).

If there is more than one rule for a specific port, we apply the most permissive rule. For example, if you have a rule that allows access to TCP port 22 (SSH) from IP address `203.0.113.1` and another rule that allows access to TCP port 22 from everyone, everyone has access to TCP port 22.

When you associate multiple security groups with an instance, the rules from each security group are effectively aggregated to create one set of rules. We use this set of rules to determine whether to allow access.

Caution
Because you can assign multiple security groups to an instance, an instance can have hundreds of rules that apply. This might cause problems when you access the instance. Therefore, we recommend that you condense your rules as much as possible.

For more information about IP addresses, see Amazon EC2 Instance IP Addressing (p. 480).

Default Security Groups

Your AWS account automatically has a *default security group* per region for EC2-Classic. When you create a VPC, we automatically create a default security group for the VPC. If you don't specify a different security group when you launch an instance, the instance is automatically associated with the appropriate default security group.

A default security group is named `default`, and it has an ID assigned by AWS. The following are the initial settings for each default security group:

- Allow inbound traffic only from other instances associated with the default security group
- Allow all outbound traffic from the instance

The default security group specifies itself as a source security group in its inbound rules. This is what allows instances associated with the default security group to communicate with other instances associated with the default security group.

You can change the rules for a default security group. For example, you can add an inbound rule to allow SSH connections so that specific hosts can manage the instance.

You can't delete a default security group. If you try to delete the EC2-Classic default security group, you'll get the following error: `Client.InvalidGroup.Reserved: The security group 'default' is reserved`. If you try to delete a VPC default security group, you'll get the following error: `Client.CannotDelete: the specified group: "sg-51530134" name: "default" cannot be deleted by a user`.

Custom Security Groups

If you don't want all your instances to use the default security group, you can create your own security groups and specify them when you launch your instances. You can create multiple security groups to reflect the different roles that your instances play; for example, a web server or a database server. For instructions that help you create security groups for web servers and database servers, see Recommended Security Groups in the *Amazon VPC User Guide*.

> **Note**
> In EC2-Classic, you can create up to 500 security groups in each region for each account. In EC2-VPC, you can create up to 100 security groups per VPC. The security groups for EC2-Classic do not count against the security group limit for EC2-VPC.

When you create a security group, you must provide it with a name and a description. Security group names and descriptions can be up to 255 characters in length, and are limited to the following characters:

- EC2-Classic: ASCII characters
- EC2-VPC: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

AWS assigns each security group a unique ID in the form sg-*xxxxxxxx*. The following are the initial settings for a security group that you create:

- Allow no inbound traffic
- Allow all outbound traffic

After you've created a security group, you can change its inbound rules to reflect the type of inbound traffic that you want to reach the associated instances. In EC2-VPC, you can also change its outbound rules.

To allow instances that have the same security group to communicate with each other, you must explicitly add rules for this. The following table describes the rules that you must add to your security group to enable instances in EC2-Classic to communicate with each other.

Inbound			
Source	**Protocol**	**Port Range**	**Comments**
The ID of the security group	ICMP	All	Allow inbound ICMP access from other instances associated with this security group
The ID of the security group	TCP	0 - 65535	Allow inbound TCP access from other instances associated with this security group
The ID of the security group	UDP	0 - 65535	Allow inbound UDP access from other instances associated with this security group

The following table describes the rules that you must add to your security group to enable instances in a VPC to communicate with each other.

Inbound			
Source	**Protocol**	**Port Range**	**Comments**
The ID of the security group	All	All	Allow inbound traffic from other instances associated with this security group

Creating a Security Group

To create a new security group

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Security Groups**.
3. Click **Create Security Group**.
4. Specify a name and description for the security group. For **VPC**, select **No VPC** to create a security group for EC2-Classic, or select a VPC ID to create a security group for that VPC.
5. You can start adding rules, or you can click **Create** to create the security group now (you can always add rules later). For more information about adding rules, see .

To copy a security group

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Security Groups**.
3. Select the security group you want to copy, click **Actions**, and then select **Copy to new**.
4. The **Create Security Group** dialog opens, and is populated with the rules from the existing security group. Specify a name and description for your new security group. In the **VPC** list, select **No VPC** to create a security group for EC2-Classic, or select a VPC ID to create a security group for that VPC. When you are done, click **Create**.

You can assign a security group to an instance when you launch the instance. When you add or remove rules, those changes are automatically applied to all instances to which you've assigned the security group.

After you launch an instance in EC2-Classic, you can't change its security groups. After you launch an instance in a VPC, you can change its security groups. For more information, see Changing an Instance's Security Groups in the *Amazon VPC User Guide*.

Describing Your Security Groups

To describe your security groups for EC2-Classic

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Security Groups**.
3. Select **Network Platforms** from the filter list, then select **EC2-Classic**.
4. Select a security group. We display general information in the **Description** tab and inbound rules on the **Inbound** tab.

To describe your security groups for EC2-VPC

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Security Groups**.
3. Select **Network Platforms** from the filter list, then select **EC2-VPC**.
4. Select a security group. We display general information in the **Description** tab, inbound rules on the **Inbound** tab, and outbound rules on the **Outbound** tab.

Adding Rules to a Security Group

When you add a rule to a security group, the new rule is automatically applied to any instances associated with the security group.

To add rules to a security group

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Security Groups**.
3. Select the security group.
4. You can allow web servers to receive all inbound HTTP and HTTPS traffic. On the **Inbound** tab, click **Edit**. In the dialog, click **Add Rule**. Select **HTTP** from the **Type** list, and leave the source as **Anywhere** (`0.0.0.0/0`). Add a similar rule for the HTTPS protocol.

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

In the **Source** field, specify the public IP address of your computer, in CIDR notation. For example, if your IP address is `203.0.113.25`, specify `203.0.113.25/32` to list this single IP address in CIDR notation. If your company allocates addresses from a range, specify the entire range, such as `203.0.113.0/24`. You can select **My IP** to from the **Source** list to let us automatically populate the field with your computer's IP address. However, if you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

> **Caution**
> If you use `0.0.0.0/0`, you enable all IP addresses to access your instance using SSH. This is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, you'll authorize only a specific IP address or range of addresses to access your instance.

6. You can allow communication between all instances associated with this security group, or between instances associated with another security group and instances associated with this security group. Click **Add Rule**, select **All ICMP**, then start typing the ID of the security group in **Source**; this provides you with a list of security groups. Select the security group from the list. Repeat the steps for the TCP and UDP protocols. Click **Save** when you are done.

7. If you are creating a security group for a VPC, you can also specify outbound rules. For an example, see Adding and Removing Rules in the *Amazon VPC User Guide*.

Deleting Rules from a Security Group

When you delete a rule from a security group, the change is automatically applied to any instances associated with the security group.

To delete a security group rule

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Security Groups**.
3. Select a security group.
4. Click **Edit**, and then click the **Delete** icon next to each rule that you need to delete.
5. Click **Save**.

Deleting a Security Group

You can't delete a security group that associated with an instance. You can't delete the default security group.

To delete a security group

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Security Groups**.
3. Select a security group and click **Delete**.
4. Click **Yes, Delete**.

API and Command Overview

You can perform the tasks described on this page using the command line or an API. For more information about the command line interfaces and a list of available APIs, see Accessing Amazon EC2 (p. 3).

Create a security group

*   create-security-group (AWS CLI)
*   ec2-create-group (Amazon EC2 CLI)
*   New-EC2SecurityGroup (AWS Tools for Windows PowerShell)

Add one or more ingress rules to a security group

*   authorize-security-group-ingress (AWS CLI)
*   ec2-authorize (Amazon EC2 CLI)
*   Grant-EC2SecurityGroupIngress (AWS Tools for Windows PowerShell)

[EC2-VPC] Add one or more egress rules to a security group

*   authorize-security-group-egress (AWS CLI)
*   ec2-authorize (Amazon EC2 CLI)
*   Grant-EC2SecurityGroupIngress (AWS Tools for Windows PowerShell)

Describe one or more security groups

*   describe-security-groups (AWS CLI)
*   ec2-describe-group (Amazon EC2 CLI)
*   Get-EC2SecurityGroup (AWS Tools for Windows PowerShell)

[EC2-VPC] Modify the security groups for an instance

*   modify-instance-attribute (AWS CLI)
*   ec2-modify-instance-attribute (Amazon EC2 CLI)
*   Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

Remove one or more ingress rules from a security group

- revoke-security-group-ingress (AWS CLI)
- ec2-revoke (Amazon EC2 CLI)
- Revoke-EC2SecurityGroupIngress (AWS Tools for Windows PowerShell)

[EC2-VPC] Remove one or more egress rules from a security group

- revoke-security-group-egress(AWS CLI)
- ec2-revoke (Amazon EC2 CLI)
- Revoke-EC2SecurityGroupEgress (AWS Tools for Windows PowerShell)

Delete a security group

- delete-security-group (AWS CLI)
- ec2-delete-group (Amazon EC2 CLI)
- Remove-EC2SecurityGroup (AWS Tools for Windows PowerShell)

Controlling Access to Amazon EC2 Resources

Your security credentials identify you to services in AWS and grant you unlimited use of your AWS re-
sources, such as your Amazon EC2 resources. You can use features of Amazon EC2 and AWS Identity
and Access Management (IAM) to allow other users, services, and applications to use your Amazon EC2
resources without sharing your security credentials. You can use IAM to control how other users use re-
sources in your AWS account, and you can use security groups to control access to your Amazon EC2
instances. You can choose to allow full use or limited use of your Amazon EC2 resources.

Topics
- Network Access to Your Instance (p. 429)
- Amazon EC2 Permission Attributes (p. 429)
- IAM and Amazon EC2 (p. 430)
- IAM Policies for Amazon EC2 (p. 431)
- IAM Roles for Amazon EC2 (p. 461)
- Authorizing Inbound Traffic for Your Instances (p. 467)

Network Access to Your Instance

A security group acts as a firewall that controls the traffic allowed to reach one or more instances. When
you launch an instance, you assign it one or more security groups. You add rules to each security group
that control traffic for the instance. You can modify the rules for a security group at any time; the new
rules are automatically applied to all instances to which the security group is assigned.

For more information, see Authorizing Inbound Traffic for Your Instances (p. 467).

Amazon EC2 Permission Attributes

Your organization might have multiple AWS accounts. Amazon EC2 enables you to specify additional
AWS accounts that can use your Amazon Machine Images (AMIs) and Amazon EBS snapshots. These
permissions work at the AWS account level only; you can't restrict permissions for specific users within

the specified AWS account. All users in the AWS account that you've specified can use the AMI or snapshot.

Each AMI has a `LaunchPermission` attribute that controls which AWS accounts can access the AMI. For more information, see Making an AMI Public (p. 63).

Each Amazon EBS snapshot has a `createVolumePermission` attribute that controls which AWS accounts can use the snapshot. For more information, see Sharing Snapshots (p. 563).

IAM and Amazon EC2

IAM enables you to do the following:

- Create users and groups under your AWS account
- Assign unique security credentials to each user under your AWS account
- Control each user's permissions to perform tasks using AWS resources
- Allow the users in another AWS account to share your AWS resources
- Create roles for your AWS account and define the users or services that can assume them
- Use existing identities for your enterprise to grant permissions to perform tasks using AWS resources

By using IAM with Amazon EC2, you can control whether users in your organization can perform a task using specific Amazon EC2 API actions and whether they can use specific AWS resources.

This topic helps you answer the following questions:

- How do I create groups and users in IAM?
- How do I create a policy?
- What IAM policies do I need to carry out tasks in Amazon EC2?
- How do I grant permissions to perform actions in Amazon EC2?
- How do I grant permissions to perform actions on specific resources in Amazon EC2?

Creating an IAM Group and Users

To create an IAM group

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, click **Groups** and then click **Create New Group**.
3. In the **Group Name** box, type a name for your group, and then click **Next Step**.
4. In the **Select Policy Template** section, click **Select** next to a policy template of your choice. For example, for Amazon EC2, one of the following policy templates might meet your needs:

   - Power User Access
   - Read Only Access
   - Amazon EC2 Full Access
   - Amazon EC2 Read Only Access

5. Click **Next Step** and then click **Create Group**.

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

To create an IAM user, add the user to your group, and create a password for the user

1. In the navigation pane, click **Users** and then click **Create New Users**.

2. In box **1**, type a user name and then click **Create**.

3. Click **Download Credentials** and save your access key in a secure place. You will need your access key for programmatic access to AWS using the AWS CLI, the AWS SDKs, or the HTTP APIs.

    > **Note**
    > You cannot retrieve the secret access key after you complete this step; if you misplace it you must create a new one.

    After you have downloaded your access key, click **Close**.

4. Under **User Name**, click the name of the user you just created.

5. Click **Groups** and then click **Add User to Groups**.

6. Select the group you created earlier, and then click **Add to Groups**.

7. Click **Security Credentials** and then under **Sign-In Credentials**, click **Manage Password**.

8. Select **Assign a custom password** and then type and confirm a password. When you are finished, click **Apply**.

9. Give each user his or her credentials (access keys and password); this enables them to use services based on the permissions you specified for the IAM group

Related Topics

For more information about IAM, see the following:

*   IAM Policies for Amazon EC2 (p. 431)
*   IAM Roles for Amazon EC2 (p. 461)
*   Identity and Access Management (IAM)
*   Using IAM

IAM Policies for Amazon EC2

By default, IAM users don't have permission to create or modify Amazon EC2 resources, or perform tasks using the Amazon EC2 API. (This means that they also can't do so using the Amazon EC2 console or CLI.) To allow IAM users to create or modify resources and perform tasks, you must create IAM policies that grant IAM users permission to use the specific resources and API actions they'll need, and then attach those policies to the IAM users or groups that require those permissions.

When you attach a policy to a user or group of users, it allows or denies the users permission to perform the specified tasks on the specified resources. For more general information about IAM policies, see Permissions and Policies in the *Using IAM* guide.

Getting Started

An IAM policy must grant or deny permission to use one or more Amazon EC2 actions. It must also specify the resources that can be used with the action, which can be all resources, or in some cases, specific resources. The policy can also include conditions that you apply to the resource.

Amazon EC2 partially supports resource-level permissions. This means that for some EC2 API actions, you cannot specify which resource a user is allowed to work with for that action; instead, you have to allow users to work with all resources for that action.

Task	Topic
Understand the basic structure of a policy	Policy Syntax (p. 432)
Define actions in your policy	Actions for Amazon EC2 (p. 433)
Define specific resources in your policy	Amazon Resource Names for Amazon EC2 (p. 433)
Apply conditions to the use of the resources	Condition Keys for Amazon EC2 (p. 436)
Work with the available resource-level permissions for Amazon EC2	Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 439)
Test your policy	Checking that Users Have the Required Permissions (p. 438)
Example policies for a CLI or SDK	Example Policies for Working With the AWS CLI, the Amazon EC2 CLI, or an AWS SDK (p. 446)
Example policies for the Amazon EC2 console	Example Policies for Working in the Amazon EC2 Console (p. 453)

Policy Structure

The following topics explain the structure of an IAM policy.

Topics

Policy Syntax

An IAM policy is a JSON document that consists of one of more statements. Each statement is structured as follows:

```
{
  "Statement":[{
    "Effect":"effect",
    "Action":"action",
    "Resource":"arn",
    "Condition":{
      "condition":{
        "key":"value"
        }
      }
    }
  ]
}
```

There are various elements that make up a statement:

- **Effect:** The *effect* can be `Allow` or `Deny`. By default, IAM users don't have permission to use resources and API actions, so all requests are denied. An explicit allow overrides the default. An explicit deny overrides any allows.
- **Action**: The *action* is the specific API action for which you are granting or denying permission. To learn about specifying *action*, see Actions for Amazon EC2 (p. 433).
- **Resource**: The resource that's affected by the action. Some Amazon EC2 API actions allow you to include specific resources in your policy that can be created or modified by the action. To specify a resource in the statement, you need to use its Amazon Resource Name (ARN). For more information about specifying the *arn* value, see Amazon Resource Names for Amazon EC2 (p. 433). For more information about which API actions support which ARNs, see Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 439). If the API action does not support ARNs, use the * wildcard to specify that all resources can be affected by the action.
- **Condition**: Conditions are optional. They can be used to control when your policy will be in effect. For more information about specifying conditions for Amazon EC2, see Condition Keys for Amazon EC2 (p. 436).

For more information about example IAM policy statements for Amazon EC2, see Example Policies for Working With the AWS CLI, the Amazon EC2 CLI, or an AWS SDK (p. 446).

Actions for Amazon EC2

In an IAM policy statement, you can specify any API action from any service that supports IAM. For Amazon EC2, use the following prefix with the name of the API action: `ec2:`. For example: `ec2:RunInstances` and `ec2:CreateImage`.

To specify multiple actions in a single statement, separate them with commas as follows:

```
"Action": ["ec2:action1", "ec2:action2"]
```

You can also specify multiple actions using wildcards. For example, you can specify all actions whose name begins with the word "Describe" as follows:

```
"Action": "ec2:Describe*"
```

To specify all Amazon EC2 API actions, use the * wildcard as follows:

```
"Action": "ec2:*"
```

For a list of Amazon EC2 actions, see Actions in the *Amazon EC2 API Reference*.

Amazon Resource Names for Amazon EC2

Each IAM policy statement applies to the resources that you specify using their ARNs.

Important
Currently, not all API actions support individual ARNs; we'll add support for additional API actions and ARNs for additional Amazon EC2 resources later. For information about which ARNs you can use with which Amazon EC2 API actions, as well as supported condition keys for each ARN, see Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 439).

An ARN has the following general syntax:

```
arn:aws:[service]:[region]:[account]:resourceType/resourcePath
```

service
> The service (for example, `ec2`).

region
> The region for the resource (for example, `us-east-1`).

account
> The AWS account ID, with no hyphens (for example, `123456789012`).

resourceType
> The type of resource (for example, `instance`).

resourcePath
> A path that identifies the resource. You can use the * wildcard in your paths.

For example, you can indicate a specific instance (`i-1a2b3c4d`) in your statement using its ARN as follows:

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/i-1a2b3c4d"
```

You can also specify all instances that belong to a specific account by using the * wildcard as follows:

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*"
```

To specify all resources, or if a specific API action does not support ARNs, use the * wildcard in the `Re-source` element as follows:

```
"Resource": "*"
```

The following table describes the ARNs for each type of resource used by the Amazon EC2 API actions.

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

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

Many Amazon EC2 API actions involve multiple resources. For example, `AttachVolume` attaches an Amazon EBS volume to an instance, so an IAM user must have permission to use the volume and the instance. To specify multiple resources in a single statement, separate their ARNs with commas, as follows:

```
"Resource": ["arn1", "arn2"]
```

For more general information about ARNs, see Amazon Resource Names (ARN) and AWS Service Namespaces in the *Amazon Web Services General Reference*. For more information about the resources

that are created or modified by the Amazon EC2 actions, and the ARNs that you can use in your IAM policy statements, see Granting IAM Users Required Permissions for Amazon EC2 Resources in the *Amazon EC2 API Reference*.

Condition Keys for Amazon EC2

In a policy statement, you can optionally specify conditions that control when it is in effect. Each condition contains one or more key-value pairs. Condition keys are not case sensitive. We've defined AWS-wide condition keys, plus additional service-specific condition keys.

If you specify multiple conditions, or multiple keys in a single condition, we evaluate them using a logical AND operation. If you specify a single condition with multiple values for one key, we evaluate the condition using a logical OR operation. For permission to be granted, all conditions must be met.

You can also use placeholders when you specify conditions. For example, you can grant an IAM user permission to use resources with a tag that specifies his or her IAM user name. For more information, see Policy Variables in the *Using IAM* guide.

Amazon EC2 implements the AWS-wide condition keys (see Available Keys), plus the following service-specific condition keys. (We'll add support for additional service-specific condition keys for Amazon EC2 later.)

Condition Key	Key/Value Pair	Evaluation Types
ec2:AccepterVpc	"ec2:AccepterVpc":"*vpc-arn*" Where *vpc-arn* is the VPC ARN for the peer VPC	ARN, Null
ec2:AvailabilityZone	"ec2:AvailabilityZone":"*az-api-name*" Where *az-api-name* is the name of the Availability Zone (for example, `us-west-2a`) To list your Availability Zones, use ec2-describe-availability-zones	String, Null
ec2:EbsOptimized	"ec2:EbsOptimized":"*optimized-flag*" Where *optimized-flag* is `true	false`	Boolean, Null
ec2:ImageType	"ec2:ImageType":"*image-type-api-name*" Where *image-type-api-name* is `ami	aki	ari`	String, Null
ec2:InstanceProfile	"ec2:InstanceProfile":"*instance-profile-arn*" Where *instance-profile-arn* is the instance profile ARN	ARN, Null

Condition Key	Key/Value Pair	Evaluation Types
ec2:InstanceType	"ec2:InstanceType":"*instance-type-api-name*" Where *instance-type-api-name* is the name of the instance type (`t2.micro` | `t2.small` | `t2.medium` | `m3.medium` | `m3.large` | `m3.xlarge` | `m3.2xlarge` | `m1.small` | `m1.medium` | `m1.large` | `m1.xlarge` | `c3.large` | `c3.xlarge` | `c3.2xlarge` | `c3.4xlarge` | `c3.8xlarge` | `c1.medium` | `c1.xlarge` | `cc2.8xlarge` | `r3.large` | `r3.xlarge` | `r3.2xlarge` | `r3.4xlarge` | `r3.8xlarge` | `m2.xlarge` | `m2.2xlarge` | `m2.4xlarge` | `cr1.8xlarge` | `i2.xlarge` | `i2.2xlarge` | `i2.4xlarge` | `i2.8xlarge` | `hs1.8xlarge` | `hi1.4xlarge` | `t1.micro` | `g2.2xlarge` | `cg1.4xlarge`).	String, Null
ec2:Owner	"ec2:Owner":"*account-id*" Where *account-id* is `amazon` | *aws-account-id*	String, Null
ec2:ParentSnapshot	"ec2:ParentSnapshot":"*snapshot-arn*" Where *snapshot-arn* is the snapshot ARN	ARN, Null
ec2:ParentVolume	"ec2:ParentVolume":"*volume-arn*" Where *volume-arn* is the volume ARN	ARN, Null
ec2:PlacementGroup	"ec2:PlacementGroup":"*placement-group-arn*" Where *placement-group-arn* is the placement group ARN	ARN, Null
ec2:PlacementGroupStrategy	"ec2:PlacementGroupStrategy":"*placement-group-strategy*" Where *placement-group-strategy* is `cluster`	String, Null
ec2:Public	"ec2:Public":"*public-flag*" Where *public-flag* for an AMI is `true` | `false`	Boolean, Null
ec2:Region	"ec2:Region":"*region-name*" Where *region-name* is the name of the region (for example, `us-west-2`). To list your regions, use ec2-describe-regions.	String, Null
ec2:RequesterVpc	"ec2:RequesterVpc":"*vpc-arn*" Where *vpc-arn* is the VPC ARN for the requester's VPC	ARN, Null
ec2:ResourceTag/*tag-key*	"ec2:ResourceTag/*tag-key*":"*tag-value*" Where *tag-key* and *tag-value* are the tag-key pair	String, Null
ec2:RootDeviceType	"ec2:RootDeviceType":"*root-device-type-name*" Where *root-device-type-name* is `ebs` | `instance-store`	String, Null
ec2:Subnet	"ec2:Subnet":"*subnet-arn*" Where *subnet-arn* is the subnet ARN	ARN, Null

Condition Key	Key/Value Pair	Evaluation Types
ec2:Tenancy	"ec2:Tenancy":"*tenancy-attribute*" Where *tenancy-attribute* is `default	dedicated`	String, Null
ec2:VolumeIops	"ec2:VolumeIops":"*volume-iops*" Where *volume-iops* is the input/output operations per second (IOPS); the range is 100 to 4,000	Numeric, Null
ec2:VolumeSize	"ec2:VolumeSize":"*volume-size*" Where *volume-size* is the size of the volume, in GiB	Numeric, Null
ec2:VolumeType	"ec2:VolumeType":"*volume-type-name*" Where *volume-type-name* is `gp2` for General Purpose (SSD) volumes, `standard` for Magnetic Amazon EBS volumes, or `io1` for Provisioned IOPS (SSD) volumes.	String, Null
ec2:Vpc	"ec2:Vpc":"*vpc-arn*" Where *vpc-arn* is the VPC ARN	ARN, Null

For information about which condition keys you can use with which Amazon EC2 resources, on an action-by-action basis, see Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 439). For example policy statements for Amazon EC2, see Example Policies for Working With the AWS CLI, the Amazon EC2 CLI, or an AWS SDK (p. 446).

Checking that Users Have the Required Permissions

After you've created an IAM policy, we recommend that you check whether it grants users the permissions to use the particular API actions and resources they need before you put the policy into production.

First, create an IAM user for testing purposes, and then attach the IAM policy that you created to the test user. Then, make a request as the test user.

If the action that you are testing creates or modifies a resource, you should make the request using the `DryRun` parameter (or run the CLI command with the `--auth-dry-run` option). In this case, the call completes the authorization check, but does not complete the operation. For example, you can check whether the user can terminate a particular instance without actually terminating it. If the test user has the required permissions, the request returns `DryRunOperation`; otherwise, it returns `Unauthorized-Operation`.

If the policy doesn't grant the user the permissions that you expected, or is overly permissive, you can adjust the policy as needed and retest until you get the desired results.

> **Important**
> It can take several minutes for policy changes to propagate before they take effect. Therefore, we recommend that you allow five minutes to pass before you test your policy updates.

If an authorization check fails, the request returns an encoded message with diagnostic information. You can decode the message using the `DecodeAuthorizationMessage` action. For more information, see DecodeAuthorizationMessage in the *AWS Security Token Service API Reference*, and decode-authorization-message in the *AWS Command Line Interface Reference*.

For additional information about resource-level permissions in Amazon EC2, see the following AWS Security Blog post: Demystifying EC2 Resource-Level Permissions.

Supported Resource-Level Permissions for Amazon EC2 API Actions

Resource-level permissions refers to the ability to specify which resources users are allowed to perform actions on. Amazon EC2 has partial support for resource-level permissions. This means that for certain Amazon EC2 actions, you can control when users are allowed to use those actions based on conditions that have to be fulfilled, or specific resources that users are allowed to use. For example, you can grant users permission to launch instances, but only of a specific type, and only using a specific AMI.

The following table describes the Amazon EC2 API actions that currently support resource-level permissions, as well as the supported resources (and their ARNs) and condition keys for each action.

> **Important**
> If an Amazon EC2 API action is not listed in this table, then it does not support resource-level permissions. If an Amazon EC2 API action does not support resource-level permissions, you can grant users permission to use the action, but you have to specify a * for the resource element of your policy statement. For an example of how to do this, see 1: Allow users to list the Amazon EC2 resources that belong to the AWS account (p. 446). We'll add support for additional actions, ARNs, and condition keys later.

API Action	Resources	Condition Keys
AcceptVpcPeeringConnection	VPC peering connection arn:aws:ec2:*region*:*account*:vpc-peering-connection/*vpc-peering-connection-id*	ec2:AccepterVpc ec2:Region ec2:ResourceTag/*tag-key* ec2:RequesterVpc
	VPC arn:aws:ec2:*region*:*account*:vpc/*vpc-id* Where *vpc-id* is a VPC owned by the accepter.	ec2:ResourceTag/*tag-key* ec2:Region ec2:Tenancy

API Action	Resources	Condition Keys
AttachVolume	Instance arn:aws:ec2:*region*:*account*:instance/*instance-id*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/*tag-key* ec2:RootDeviceType ec2:Tenancy
	Volume arn:aws:ec2:*region*:*account*:volume/*volume-id*	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:ResourceTag/*tag-key* ec2:VolumeIops ec2:VolumeSize ec2:VolumeType
AuthorizeSecurityGroupEgress	Security group arn:aws:ec2:*region*:*account*:security-group/*security-group-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
AuthorizeSecurityGroupIngress	Security group arn:aws:ec2:*region*:*account*:security-group/*security-group-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
CreateVpcPeeringConnection	VPC arn:aws:ec2:*region*:*account*:vpc/*vpc-id* Where *vpc-id* is a requester VPC.	ec2:ResourceTag/*tag-key* ec2:Region ec2:Tenancy
	VPC peering connection arn:aws:ec2:*region*:*account*:vpc-peering-connection/*	ec2:AccepterVpc ec2:Region ec2:RequesterVpc
DeleteCustomerGateway	Customer gateway arn:aws:ec2:*region*:*account*:customer-gateway/*cgw-id*	ec2:Region ec2:ResourceTag/*tag-key*

API Action	Resources	Condition Keys
DeleteDhcpOptions	DHCP options set arn:aws:ec2:*region*:*account*:dhcp-options/*dhcp-options-id*	ec2:Region ec2:ResourceTag/*tag-key*
DeleteInternetGateway	Internet gateway arn:aws:ec2:*region*:*account*:internet-gateway/*igw-id*	ec2:Region ec2:ResourceTag/*tag-key*
DeleteNetworkAcl	Network ACL arn:aws:ec2:*region*:*account*:network-acl/*nacl-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
DeleteNetworkAclEntry	Network ACL arn:aws:ec2:*region*:*account*:network-acl/*nacl-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
DeleteRoute	Route table arn:aws:ec2:*region*:*account*:route-table/*route-table-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
DeleteRouteTable	Route table arn:aws:ec2:*region*:*account*:route-table/*route-table-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
DeleteSecurityGroup	Security group arn:aws:ec2:*region*:*account*:security-group/*security-group-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
DeleteVolume	Volume arn:aws:ec2:*region*:*account*:volume/*volume-id*	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:ResourceTag/*tag-key* ec2:VolumeIops ec2:VolumeSize ec2:VolumeType
DeleteVpcPeeringConnection	VPC peering connection arn:aws:ec2:*region*:*account*:vpc-peering-connection/*vpc-peering-connection-id*	ec2:AccepterVpc ec2:Region ec2:ResourceTag/*tag-key* ec2:RequesterVpc

API Action	Resources	Condition Keys
DetachVolume	Instance arn:aws:ec2:*region*:*account*:instance/*instance-id*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/*tag-key* ec2:RootDeviceType ec2:Tenancy
	Volume arn:aws:ec2:*region*:*account*:volume/*volume-id*	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:ResourceTag/*tag-key* ec2:VolumeIops ec2:VolumeSize ec2:VolumeType
RebootInstances	Instance arn:aws:ec2:*region*:*account*:instance/*instance-id*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/*tag-key* ec2:RootDeviceType ec2:Tenancy
RejectVpcPeeringConnection	VPC peering connection arn:aws:ec2:*region*:*account*:vpc-peering-connection/*vpc-peering-connection-id*	ec2:AccepterVpc ec2:Region ec2:ResourceTag/*tag-key* ec2:RequesterVpc

API Action	Resources	Condition Keys
RevokeSecurityGroupEgress	Security group arn:aws:ec2:*region*:*account*:security-group/*security-group-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
RevokeSecurityGroupIngress	Security group arn:aws:ec2:*region*:*account*:security-group/*security-group-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc

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

API Action	Resources	Condition Keys
		ec2:Owner
		ec2:ParentVolume
		ec2:Region
		ec2:SnapshotTime
		ec2:ResourceTag/*tag-key*
		ec2:VolumeSize
	Subnet arn:aws:ec2:*region*:*account*:subnet/*subnet-id*	ec2:AvailabilityZone ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
	Volume arn:aws:ec2:*region*:*account*:volume/*volume-id*	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:VolumeIops ec2:VolumeSize ec2:VolumeType
StartInstances	Instance arn:aws:ec2:*region*:*account*:instance/*instance-id*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/*tag-key* ec2:RootDeviceType ec2:Tenancy

API Action	Resources	Condition Keys
StopInstances	Instance arn:aws:ec2:*region*:*account*:instance/*instance-id*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/*tag-key* ec2:RootDeviceType ec2:Tenancy
TerminateInstances	Instance arn:aws:ec2:*region*:*account*:instance/*instance-id*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/*tag-key* ec2:RootDeviceType ec2:Tenancy

Example Policies for Working With the AWS CLI, the Amazon EC2 CLI, or an AWS SDK

The following examples show policy statements that you could use to control the permissions that IAM users have to Amazon EC2. These policies are designed for requests that are made with the AWS CLI, the Amazon EC2 CLI, or an AWS SDK. For example policies for working in the Amazon EC2 console, see Example Policies for Working in the Amazon EC2 Console (p. 453). For examples of IAM policies specific to Amazon VPC, see Controlling Access to Amazon VPC Resources

Example 1: Allow users to list the Amazon EC2 resources that belong to the AWS account

The following policy grants users permission to use all Amazon EC2 API actions whose names begin with `Describe`. The `Resource` element uses a wildcard to indicate that users can specify all resources

with these API actions. The * wildcard is also necessary in cases where the API action does not support resource-level permissions. For more information about which ARNs you can use with which Amazon EC2 API actions, see Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 439).

Users don't have permission to perform any actions on the resources (unless another statement grants them permission to do so) because they're denied permission to use API actions by default.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    }
   ]
}
```

Example 2: Allow users to describe, launch, stop, start, and terminate all instances

The following policy grants users permission to use the API actions specified in the `Action` element. The `Resource` element uses a * wildcard to indicate that users can specify all resources with these API actions. The * wildcard is also necessary in cases where the API action does not support resource-level permissions. For more information about which ARNs you can use with which Amazon EC2 API actions, see Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 439).

The users don't have permission to use any other API actions (unless another statement grants them permission to do so) because users are denied permission to use API actions by default.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances", "ec2:DescribeImages",
        "ec2:DescribeKeyPairs", "ec2:DescribeSecurityGroups",
        "ec2:DescribeAvailabilityZones",
        "ec2:RunInstances", "ec2:TerminateInstances",
        "ec2:StopInstances", "ec2:StartInstances"
      ],
      "Resource": "*"
    }
   ]
}
```

Example 3: Allow users to describe all instances, and stop, start, and terminate only particular instances

The following policy allows users to describe all instances, to start and stop only instances i-123abc12 and i-4c3b2a1, and to terminate only instances in the US East (N. Virginia) region (`us-east-1`) with the resource tag "`purpose=test`".

The first statement uses a * wildcard for the `Resource` element to indicate that users can specify all resources with the action; in this case, they can list all instances. The * wildcard is also necessary in cases where the API action does not support resource-level permissions (in this case, `ec2:DescribeInstances`). For more information about which ARNs you can use with which Amazon EC2 API actions, see Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 439).

The second statement uses resource-level permissions for the `StopInstances` and `StartInstances` actions. The specific instances are indicated by their ARNs in the `Resource` element.

The third statement allows users to terminate all instances in the US East (N. Virginia) region (`us-east-1`) that belong to the specified AWS account, but only where the instance has the tag `"purpose=test"`. The `Condition` element qualifies when the policy statement is in effect.

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
   "Effect": "Allow",
      "Action": "ec2:DescribeInstances",
      "Resource": "*"
   },
   {
      "Effect": "Allow",
      "Action": [
        "ec2:StopInstances",
        "ec2:StartInstances"
      ],
      "Resource": [
      "arn:aws:ec2:us-east-1:123456789012:instance/i-123abc12",
      "arn:aws:ec2:us-east-1:123456789012:instance/i-4c3b2a1"
      ]
   },
   {
      "Effect": "Allow",
      "Action": "ec2:TerminateInstances",
      "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
      "Condition": {
         "StringEquals": {
            "ec2:ResourceTag/purpose": "test"
         }
      }
   }

   ]
}
```

Example 4. Allow users to manage particular volumes for particular instances

When an API action requires a caller to specify multiple resources, you must create a policy statement that allows users to access all required resources. If you need to use a `Condition` element with one or more of these resources, you must create multiple statements as shown in this example.

The following policy allows users to attach volumes with the tag "`volume_user=`*iam-user-name*" to instances with the tag "`department=dev`", and to detach those volumes from those instances. If you attach this policy to an IAM group, the `aws:username` policy variable gives each IAM user in the group permission to attach or detach volumes from the instances with a tag named `volume_user` that has his or her IAM user name as a value.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
```

```
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/department": "dev"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/volume_user": "${aws:username}"
        }
      }
    }
  ]
}
```

Example 5: Allow users to launch instances with a specific configuration

The RunInstances API action launches one or more instances. `RunInstances` requires an AMI and creates an instance; and users can specify a key pair and security group in the request. Launching into EC2-VPC requires a subnet, and creates a network interface. Launching from an Amazon EBS-backed AMI creates a volume. Therefore, the user must have permission to use these Amazon EC2 resources. The caller can also configure the instance using optional parameters to `RunInstances`, such as the instance type and a subnet. You can create a policy statement that requires users to specify an optional parameter, or restricts users to particular values for a parameter. The examples in this section demonstrate some of the many possible ways that you can control the configuration of an instance that a user can launch.

Note that by default, users don't have permission to describe, start, stop, or terminate the resulting instances. One way to grant the users permission to manage the resulting instances is to create a specific tag for each instance, and then create a statement that enables them to manage instances with that tag. For more information, see Example 3: Allow users to stop and start only particular instances (p. 447).

a. AMI

The following policy allows users to launch instances using only the AMIs that have the specified tag, "`department=dev`", associated with them. The users can't launch instances using other AMIs because the `Condition` element of the first statement requires that users specify an AMI that has this tag. The users also can't launch into a subnet, as the policy does not grant permissions for the subnet and network interface resources. They can, however, launch into EC2-Classic. The second statement uses a wildcard to enable users to create instance resources, and requires users to specify the key pair `project_keypair` and the security group `sg-1a2b3c4d`. Users are still able to launch instances without a key pair.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
```

```
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region::image/ami-*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/department": "dev"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region:account:instance/*",
                "arn:aws:ec2:region:account:volume/*",
                "arn:aws:ec2:region:account:key-pair/project_keypair",
                "arn:aws:ec2:region:account:security-group/sg-1a2b3c4d"
            ]
        }
    ]
}
```

Alternatively, the following policy allows users to launch instances using only the specified AMIs, `ami-9e1670f7` and `ami-45cf5c3c`. The users can't launch an instance using other AMIs (unless another statement grants the users permission to do so), and the users can't launch an instance into a subnet.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "ec2:RunInstances",
        "Resource": [
          "arn:aws:ec2:region::image/ami-9e1670f7",
          "arn:aws:ec2:region::image/ami-45cf5c3c",
          "arn:aws:ec2:region:account:instance/*",
          "arn:aws:ec2:region:account:volume/*",
          "arn:aws:ec2:region:account:key-pair/*",
          "arn:aws:ec2:region:account:security-group/*"
        ]
      }
    ]
}
```

Alternatively, the following policy allows users to launch instances from all AMIs owned by Amazon. The `Condition` element of the first statement tests whether `ec2:Owner` is `amazon`. The users can't launch an instance using other AMIs (unless another statement grants the users permission to do so). The users are able to launch an instance into a subnet.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "ec2:RunInstances",
        "Resource": [
            "arn:aws:ec2:region::image/ami-*"
```

```
        ],
        "Condition": {
            "StringEquals": {
                "ec2:Owner": "amazon"
                }
            }
    },
    {
        "Effect": "Allow",
        "Action": "ec2:RunInstances",
        "Resource": [
            "arn:aws:ec2:region:account:instance/*",
            "arn:aws:ec2:region:account:subnet/*",
            "arn:aws:ec2:region:account:volume/*",
            "arn:aws:ec2:region:account:network-interface/*",
            "arn:aws:ec2:region:account:key-pair/*",
            "arn:aws:ec2:region:account:security-group/*"
            ]
    }
    ]
}
```

b. Instance type

The following policy allows users to launch instances using only the `t1.micro` or `m1.small` instance type, which you might do to control costs. The users can't launch larger instances because the `Condition` element of the first statement tests whether `ec2:InstanceType` is either `t1.micro` or `m1.small`.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "ec2:RunInstances",
        "Resource": [
            "arn:aws:ec2:region:account:instance/*"
        ],
        "Condition": {
            "StringEquals": {
                "ec2:InstanceType": ["t1.micro", "m1.small"]
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": "ec2:RunInstances",
        "Resource": [
            "arn:aws:ec2:region::image/ami-*",
            "arn:aws:ec2:region:account:subnet/*",
            "arn:aws:ec2:region:account:network-interface/*",
            "arn:aws:ec2:region:account:volume/*",
            "arn:aws:ec2:region:account:key-pair/*",
            "arn:aws:ec2:region:account:security-group/*"
            ]
    }
    ]
}
```

c. Subnet

The following policy allows users to launch instances using only the specified subnet, `subnet-12345678`. The group can't launch instances into any another subnet (unless another statement grants the users permission to do so). Users are still able to launch instances into EC2-Classic.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region:account:subnet/subnet-12345678",
        "arn:aws:ec2:region:account:network-interface/*",
        "arn:aws:ec2:region:account:instance/*",
        "arn:aws:ec2:region:account:volume/*",
        "arn:aws:ec2:region::image/ami-*",
        "arn:aws:ec2:region:account:key-pair/*",
        "arn:aws:ec2:region:account:security-group/*"
      ]
   }
   ]
}
```

Alternatively, you could create a policy that denies users permission to launch an instance into any other subnet. The statement does this by denying permission to create a network interface, except where subnet `subnet-12345678` is specified. This denial overrides any other policies that are created to allow launching instances into other subnets. Users are still able to launch instances into EC2-Classic.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:region:account:network-interface/*"
      ],
      "Condition": {
         "ArnNotEquals": {
            "ec2:Subnet": "arn:aws:ec2:region:account:subnet/subnet-12345678"
            }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:region::image/ami-*",
         "arn:aws:ec2:region:account:network-interface/*",
         "arn:aws:ec2:region:account:instance/*",
         "arn:aws:ec2:region:account:subnet/*",
         "arn:aws:ec2:region:account:volume/*",
         "arn:aws:ec2:region:account:key-pair/*",
         "arn:aws:ec2:region:account:security-group/*"
         ]
   }
```

```
      ]
}
```

Example Policies for Working in the Amazon EC2 Console

You can use IAM policies to grant users permissions to view and work with specific resources in the Amazon EC2 console. You can use the example policies in the previous section; however, they are designed for requests that are made with the AWS CLI, the Amazon EC2 CLI, or an AWS SDK. The console uses additional API actions for its features, so these policies may not work as expected. For example, a user that has permission to use only the `DescribeVolumes` API action will encounter errors when trying to view volumes in the console. This section demonstrates policies that enable users to work with specific parts of the console.

> **Note**
> To help you work out which API actions are required to perform tasks in the console, you can use a service such as AWS CloudTrail. For more information, see the AWS CloudTrail User Guide. If your policy does not grant permission to create or modify a specific resource, the console displays an encoded message with diagnostic information. You can decode the message using the DecodeAuthorizationMessage API action for AWS STS, or the decode-authorization-message command in the AWS CLI.

For additional information about creating policies for the Amazon EC2 console, see the following AWS Security Blog post: Granting Users Permission to Work in the Amazon EC2 Console.

Example 1: Read-only access

To allow users to view all resources in the Amazon EC2 console, you can use the same policy as the following example: 1: Allow users to list the Amazon EC2 resources that belong to the AWS account (p. 446). Users cannot perform any actions on those resources or create new resources, unless another statement grants them permission to do so.

Alternatively, you can provide read-only access to a subset of resources. To do this, replace the * wildcard in the `ec2:Describe` API action with specific `ec2:Describe` actions for each resource. The following policy allows users to view all instances, AMIs, and snapshots in the Amazon EC2 console. The `ec2:DescribeTags` action allows users to view public AMIs; you can remove this action if you want users to view only private AMIs.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeInstances", "ec2:DescribeImages",
            "ec2:DescribeTags", "ec2:DescribeSnapshots"
        ],
        "Resource": "*"
    }
    ]
}
```

Note

Currently, the Amazon EC2 `ec2:Describe*` API actions do not support resource-level permissions, so you cannot control which individual resources users can view in the console. Therefore, the * wildcard is necessary in the `Resource` element of the above statement. For more information about which ARNs you can use with which Amazon EC2 API actions, see Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 439).

Example 2: Using the EC2 launch wizard

The Amazon EC2 launch wizard is a series of screens with options to configure and launch an instance. Your policy must include permission to use the API actions that allow users to work with the wizard's options. If your policy does not include permission to use those actions, some items in the wizard cannot load properly, and users cannot complete a launch.

To complete a launch successfully, users must be given permission to use the `ec2:RunInstances` API action, and at least the following API actions:

- `ec2:DescribeImages`: To view and select an AMI.
- `ec2:DescribeVPCs`: To view the available network options, which are EC2-Classic and a list of VPCs. This is required even if you are not launching into a VPC.
- `ec2:DescribeSubnets`: If launching into a VPC, to view all available subnets for the chosen VPC.
- `ec2:DescribeSecurityGroups`: To view the security groups page in the wizard. Users can select an existing security group.
- `ec2:DescribeKeyPairs` or `ec2:CreateKeyPair`: To select an existing key pair, or create a new one.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
 "ec2:DescribeInstances", "ec2:DescribeImages",
       "ec2:DescribeKeyPairs","ec2:DescribeVpcs", "ec2:DescribeSubnets",
       "ec2:DescribeSecurityGroups"
      ],
      "Resource": "*"
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*"
   }
   ]
}
```

You can add API actions to your policy to provide more options for users, for example:

- `ec2:DescribeAvailabilityZones`: If launching into EC2-Classic, to view and select a specific Availability Zone.
- `ec2:DescribeNetworkInterfaces`: If launching into a VPC, to view and select existing network interfaces for the selected subnet.
- `ec2:CreateSecurityGroup`: To create a new security group; for example, to create the wizard's suggested `launch-wizard-x` security group. However, this action alone only creates the security group; it does not add or modify any rules. To add inbound rules, users must be granted permission to use the `ec2:AuthorizeSecurityGroupIngress` API action. To add outbound rules to VPC security groups, users must be granted permission to use the `ec2:AuthorizeSecurityGroupEgress` API action. To modify or delete existing rules, users must be granted permission to use the relevant `ec2:RevokeSecurityGroup*` API action.
- `ec2:CreateTags`: To add a tag to the instance. By default, the launch wizard attempts to add a tag with a key of `Name` to an instance. Users that do not have permission to use this action will encounter a warning that this tag could not be applied to an instance; however, this does not affect the success of the launch, so you should only grant users permission to use this action if it's absolutely necessary.

Important
Be careful about granting users permission to use the `ec2:CreateTags` action. This limits your ability to use the `ec2:ResourceTag` condition key to restrict the use of other resources; users can change a resource's tag in order to bypass those restrictions.

Currently, the Amazon EC2 `Describe*` API actions do not support resource-level permissions, so you cannot restrict which individual resources users can view in the launch wizard. However, you can apply resource-level permissions on the `ec2:RunInstances` API action to restrict which resources users can use to launch an instance. The launch fails if users select options that they are not authorized to use.

The following policy allows users to launch `m1.small` instances using AMIs owned by Amazon, and only into a specific subnet (`subnet-1a2b3c4d`). Users can only launch in the sa-east-1 region. If users select a different region, or select a different instance type, AMI, or subnet in the launch wizard, the launch fails.

The first statement grants users permission to view the options in the launch wizard, as demonstrated in the example above. The second statement grants users permission to use the network interface, volume, key pair, security group, and subnet resources for the `ec2:RunInstances` action, which are required to launch an instance into a VPC. For more information about using the `ec2:RunInstances` action, see . The third and fourth statements grant users permission to use the instance and AMI resources respectively, but only if the instance is an `m1.small` instance, and only if the AMI is owned by Amazon.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeInstances", "ec2:DescribeImages",
            "ec2:DescribeKeyPairs","ec2:DescribeVpcs", "ec2:DescribeSubnets",
"ec2:DescribeSecurityGroups"
    ],
    "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action":"ec2:RunInstances",
        "Resource": [
            "arn:aws:ec2:sa-east-1:111122223333:network-interface/*",
            "arn:aws:ec2:sa-east-1:111122223333:volume/*",
            "arn:aws:ec2:sa-east-1:111122223333:key-pair/*",
            "arn:aws:ec2:sa-east-1:111122223333:security-group/*",
            "arn:aws:ec2:sa-east-1:111122223333:subnet/subnet-1a2b3c4d"
        ]
    },
    {
        "Effect": "Allow",
        "Action": "ec2:RunInstances",
        "Resource": [
            "arn:aws:ec2:sa-east-1:111122223333:instance/*"
        ],
        "Condition": {
            "StringEquals": {
                "ec2:InstanceType": "m1.small"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": "ec2:RunInstances",
        "Resource": [
                "arn:aws:ec2:sa-east-1::image/ami-*"
        ],
        "Condition": {
            "StringEquals": {
                "ec2:Owner": "amazon"
            }
        }
    }
    ]
}
```

Example 3: Working with volumes

The following policy grants users permission to view and create volumes, and attach and detach volumes to specific instances.

Users can attach any volume to instances that have the tag "`purpose=test`", and also detach volumes from those instances. To attach a volume using the Amazon EC2 console, it is helpful for users to have permission to use the `ec2:DescribeInstances` action, as this allows them to select an instance from a pre-populated list in the **Attach Volume** dialog box. However, this also allows users to view all instances on the **Instances** page in the console, so you can omit this action.

In the first statement, the `ec2:DescribeVolumeStatus` and `ec2:DescribeAvailabilityZones` actions are necessary to ensure that volumes display correctly in the console.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVolumes", "ec2:DescribeVolumeStatus",
 "ec2:DescribeAvailabilityZones", "ec2:CreateVolume",
 "ec2:DescribeInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Resource": "arn:aws:ec2:region:111122223333:instance/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/purpose": "test"
        }
      }
    }
   },
   {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Resource": "arn:aws:ec2:region:111122223333:volume/*"
 }
   ]
}
```

Example 4: Working with security groups

The following policy grants users permission to view security groups in the Amazon EC2 console, and to add and remove inbound and outbound rules for existing security groups that have the tag `Depart-ment=Test`.

> **Note**
> You can't modify outbound rules for EC2-Classic security groups. For more information about security groups, see Amazon EC2 Security Groups (p. 421).

In the first statement, the `ec2:DescribeTags` action allows users to view tags in the console, which makes it easier for users to identify the security groups that they are allowed to modify.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeSecurityGroups", "ec2:DescribeTags"
        ],
        "Resource": "*"
     },
     {
        "Effect": "Allow",
        "Action": [
          "ec2:AuthorizeSecurityGroupIngress", "ec2:RevokeSecurityGroupIngress",

          "ec2:AuthorizeSecurityGroupEgress", "ec2:RevokeSecurityGroupEgress"
        ],
        "Resource": [
            "arn:aws:ec2:region:111122223333:security-group/*"
        ],
        "Condition": {
            "StringEquals": {
                "ec2:ResourceTag/Department": "Test"
            }
        }
    }
    ]
}
```

You can create a policy that allows users to work with the **Create Security Group** dialog box in the Amazon EC2 console. To use this dialog box, users must be granted permission to use at the least the following API actions:

- `ec2:CreateSecurityGroup`: To create a new security group.
- `ec2:DescribeVpcs`: To view a list of existing VPCs in the **VPC** list. This action is required even if you are not creating a security group for a VPC.

With these permissions, users can create a new security group successfully, but they cannot add any rules to it. To work with rules in the **Create Security Group** dialog box, you can add the following API actions to your policy:

- `ec2:AuthorizeSecurityGroupIngress`: To add inbound rules.
- `ec2:AuthorizeSecurityGroupEgress`: To add outbound rules to VPC security groups.
- `ec2:RevokeSecurityGroupIngress`: To modify or delete existing inbound rules. This is useful if you want to allow users to use the **Copy to new** feature in the console. This feature opens the **Create Security Group** dialog box and populates it with the same rules as the security group that was selected.

- `ec2:RevokeSecurityGroupEgress`: To modify or delete outbound rules for VPC security groups. This is useful to allow users to modify or delete the default outbound rule that allows all outbound traffic.
- `ec2:DeleteSecurityGroup`: To cater for scenarios where invalid rules cannot be saved. If a user creates a security group with an invalid rule, the console first creates the security group, then attempts to add the rules to it. After that fails, the security group is deleted. The user remains in the **Create Security Group** dialog box, where an error is displayed. The rules remain listed, so the user can correct the invalid rule and try to create the security group again. This API action is not required, but if a user is not granted permission to use it and attempts to create a security group with invalid rules, the security group is created without any rules, and the user must add them afterward.

Currently, the `ec2:CreateSecurityGroup` API action does not support resource-level permissions; however, you can apply resource-level permissions to the `ec2:AuthorizeSecurityGroupIngress` and `ec2:AuthorizeSecurityGroupEgress` actions to control how users can create rules.

The following policy grants users permission to use the **Create Security Group** dialog box, and to create inbound and outbound rules for security groups that are associated with a specific VPC (`vpc-1a2b3c4d`). Users can create security groups for EC2-Classic or another VPC, but they cannot add any rules to them. Similarly, users cannot add any rules to any existing security group that's not associated with VPC `vpc-1a2b3c4d`. Users are also granted permission to view all security groups in the console. This makes it easier for users to identify the security groups to which they can add inbound rules.

This policy also grants users permission to delete security groups that are associated with VPC `vpc-1a2b3c4d`.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
          "ec2:DescribeSecurityGroups", "ec2:CreateSecurityGroup", "ec2:De
scribeVpcs"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
          "ec2:DeleteSecurityGroup", "ec2:AuthorizeSecurityGroupIngress",
"ec2:AuthorizeSecurityGroupEgress"
        ],
        "Resource": "arn:aws:ec2:region:111122223333:security-group/*",
        "Condition":{
            "ArnEquals": {
                "ec2:Vpc": "arn:aws:ec2:region:111122223333:vpc/vpc-1a2b3c4d"
            }
        }
    }
  ]
}
```

Example 5: Working with Elastic IP addresses

The following policy grants users permission to view Elastic IP addresses in the Amazon EC2 console. The console uses the `ec2:DescribeInstances` action to display information about instances with which the Elastic IP addresses are associated. If users are not granted permission to use this action, the Elastic IP addresses page cannot load properly.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeAddresses", "ec2:DescribeInstances"
      ],
      "Resource": "*"
   }
   ]
}
```

To allow users to work with Elastic IP addresses, you can add the following actions to your policy

- `ec2:AllocateAddress`: To allocate an address for use in VPC or EC2-Classic.
- `ec2:ReleaseAddress`: To release an Elastic IP address.
- `ec2:DescribeNetworkInterfaces`: To work with the **Associate Address** dialog box. The dialog box displays the available network interfaces to which you can associate an Elastic IP address, and will not open if users are not granted permission to use this action. However, this only applies to EC2-VPC; this action is not required for associating an Elastic IP address to an instance in EC2-Classic.
- `ec2:AssociateAddress`: To associate an Elastic IP address with an instance or a network interface.
- `ec2:DisassociateAddress`: To disassociate an Elastic IP address from an instance or a network interface,

IAM Roles for Amazon EC2

Applications must sign their API requests with AWS credentials. Therefore, if you are an application developer, you need a strategy for managing credentials for your applications that run on EC2 instances. For example, you can securely distribute your AWS credentials to the instances, enabling the applications on those instances to use your credentials to sign requests, while protecting them from other users. However, it's challenging to securely distribute credentials to each instance, especially those that AWS creates on your behalf, such as Spot Instances or instances in Auto Scaling groups. You must also be able to update the credentials on each instance when you rotate your AWS credentials.

We designed IAM roles so that your applications can securely make API requests from your instances, without requiring you to manage the security credentials that the applications use. Instead of creating and distributing your AWS credentials, you can delegate permission to make API requests using IAM roles as follows:

1. Create an IAM role.
2. Define which accounts or AWS services can assume the role.
3. Define which API actions and resources the application can use after assuming the role.
4. Specify the role when you launch your instances.
5. Have the application retrieve a set of temporary credentials and use them.

For example, you can use IAM roles to grant permissions to applications running on your instances that needs to use a bucket in Amazon S3.

> **Note**
>
> Amazon EC2 uses an *instance profile* as a container for an IAM role. When you create an IAM role using the console, the console creates an instance profile automatically and gives it the same name as the role it corresponds to. If you use the AWS CLI, API, or an AWS SDK to create a role, you create the role and instance profile as separate actions, and you might give them different names. To launch an instance with an IAM role, you specify the name of its instance profile. When you launch an instance using the Amazon EC2 console, you can select a role to associate with the instance; however, the list that's displayed is actually a list of instance profile names. For more information, see Instance Profiles in the *Using IAM*.

You can specify permissions for IAM roles by creating a policy in JSON format. These are similar to the policies that you create for IAM users. If you make a change to a role, the change is propagated to all instances, simplifying credential management.

> **Note**
>
> You can't assign a role to an existing instance; you can only specify a role when you launch a new instance.

For more information about creating and using IAM roles, see Roles in the *Using IAM* guide.

Topics

Retrieving Security Credentials from Instance Metadata

An application on the instance retrieves the security credentials provided by the role from the instance metadata item `iam/security-credentials/`*role-name*. The application is granted the permissions for the actions and resources that you've defined for the role through the security credentials associated with the role. These security credentials are temporary and we rotate them automatically. We make new credentials available at least five minutes prior to the expiration of the old credentials.

> **Warning**
>
> If you use services that use instance metadata with IAM roles, ensure that you don't expose your credentials when the services make HTTP calls on your behalf. The types of services that could expose your credentials include HTTP proxies, HTML/CSS validator services, and XML processors that support XML inclusion.

The following command retrieves the security credentials for an IAM role named `s3access`.

```
$ curl http://169.254.169.254/latest/meta-data/iam/security-credentials/s3access
```

The following is example output.

```
{
  "Code" : "Success",
  "LastUpdated" : "2012-04-26T16:39:16Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "AKIAIOSFODNN7EXAMPLE",
  "SecretAccessKey" : "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
```

```
    "Token" : "token",
    "Expiration" : "2012-04-27T22:39:16Z"
}
```

For more information about instance metadata, see Instance Metadata and User Data (p. 256). For more information about temporary credentials, see the *Using Temporary Security Credentials*.

Granting an IAM User Permission to Launch an Instance with an IAM Role

To enable an IAM user to launch an instance with an IAM role, you must grant the user permission to pass the role to the instance.

For example, the following IAM policy grants users permission to launch an instance with the IAM role named `s3access`.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "iam:PassRole",
        "Resource": "arn:aws:iam::123456789012:role/s3access"
    }]
}
```

Alternatively, you could grant IAM users access to all your roles by specifying the resource as "*" in this policy. However, consider whether users who launch instances with your roles (ones that exist or that you'll create later on) might be granted permissions that they don't need or shouldn't have.

For more information, see Permissions Required for Using Roles with Amazon EC2 in the *Using IAM* guide.

Launching an Instance with an IAM Role Using the Console

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

> **Important**
> After you create an IAM role, it may take several seconds for the permissions to propagate. If your first attempt to launch an instance with a role fails, wait a few seconds before trying again. For more information, see Troubleshooting Working with Roles in the *Using IAM* guide.

To create an IAM role using the IAM console

1. Open the IAM console.
2. In the navigation pane, click **Roles**, and then click **Create New Role**.
3. On the **Set Role Name** page, enter a name for the role and click **Next Step**.
4. On the **Select Role Type** page, click **Select** next to **Amazon EC2**.
5. On the **Set Permissions** page, specify the policies for the group. You can select a policy template or create custom policies. For example, for Amazon EC2, one of the following policy templates might meet your needs:

    *   Power User Access
    *   Read Only Access
    *   Amazon EC2 Full Access

- Amazon EC2 Read Only Access

For more information about creating custom policies, see IAM Policies for Amazon EC2 (p. 431).

6. On the second **Set Permissions** page, you can replace the automatically generated policy name with a name of your choice. Check the details in the policy document, and click **Next Step**.

7. Review the role information, edit the role as needed, and then click **Create Role**.

To launch an instance with an IAM role

1. Open the Amazon EC2 console.

2. On the dashboard, click **Launch Instance**.

3. Select an AMI, then select an instance type and click **Next: Configure Instance Details**.

4. On the **Configure Instance Details** page, select the IAM role you created from the **IAM role** list.

> **Note**
> The **IAM role** list displays the name of the instance profile that you created when you created your IAM role. If you created your IAM role using the console, the instance profile was created for you and given the same name as the role. If you created your IAM role using the AWS CLI, API, or an AWS SDK, you may have named your instance profile differently.

5. Configure any other details, then follow the instructions through the rest of the wizard, or click **Review and Launch** to accept default settings and go directly to the **Review Instance Launch** page.

6. Review your settings, then click **Launch** to choose a key pair and launch your instance.

7. If you are using the Amazon EC2 API actions in your application, retrieve the AWS security credentials made available on the instance and use them to sign the requests. Note that the AWS SDK does this for you.

```
$ curl http://169.254.169.254/latest/meta-data/iam/security-creden
tials/role_name
```

Launching an Instance with an IAM Role Using the AWS CLI

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

> **Important**
> After you create an IAM role, it may take several seconds for the permissions to propagate. If your first attempt to launch an instance with a role fails, wait a few seconds before trying again. For more information, see Troubleshooting Working with Roles in the *Using IAM* guide.

To create an IAM role using the AWS CLI

- Create an IAM role with a policy that allows the role to use an Amazon S3 bucket.

a. Create the following trust policy and save it in a text file named `ec2-role-trust-policy.json`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": "ec2.amazonaws.com"},
      "Action": "sts:AssumeRole"
    }
```

```
    ]
}
```

b.   Create the `s3access` role. You'll specify the trust policy you created.

```
$ aws iam create-role --role-name s3access --assume-role-policy-document
 file://ec2-role-trust-policy.json
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": "sts:AssumeRole",
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "ec2.amazonaws.com"
                    }
                }
            ]
        },
        "RoleId": "AROAIIZKPBKS2LEXAMPLE",
        "CreateDate": "2013-12-12T23:46:37.247Z",
        "RoleName": "s3access",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/s3access"
    }
}
```

c.   Create an access policy and save it in a text file named `ec2-role-access-policy.json`.
     For example, this policy grants administrative permissions for Amazon S3 to applications running
     on the instance.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:*"],
      "Resource": ["*"]
    }
  ]
}
```

d.   Attach the access policy to the role.

```
$ aws iam put-role-policy --role-name s3access --policy-name S3-Permis
sions --policy-document file://ec2-role-access-policy.json
```

e.   Create an instance profile named `s3access-profile`.

```
$ aws iam create-instance-profile --instance-profile-name S3-Permissions
{
```

```
    "InstanceProfile": {
        "InstanceProfileId": "AIPAJTLBPJLEGREXAMPLE",
        "Roles": [],
        "CreateDate": "2013-12-12T23:53:34.093Z",
        "InstanceProfileName": "S3-Permissions",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:instance-profile/S3-Permissions"

    }
}
```

f.   Add the `s3access` role to the `s3access-profile` instance profile.

```
$ aws iam add-role-to-instance-profile --instance-profile-name S3-Permis
sions --role-name s3access
```

For more information about these commands, see create-role, put-role-policy, and create-instance-profile in the *AWS Command Line Interface Reference*.

To launch an instance with an IAM role using the AWS CLI

1.   Launch an instance using the instance profile. The following example shows how to launch an instance with the instance profile.

```
$ aws ec2 run-instances --image-id ami-11aa22bb --iam-instance-profile
Name="S3-Permissions" --key-name my-key-pair --security-groups my-security-
group --subnet-id subnet-1a2b3c4d
```

For more information, see run-instances in the *AWS Command Line Interface Reference.*

2.   If you are using the Amazon EC2 API actions in your application, retrieve the AWS security credentials made available on the instance and use them to sign the requests. Note that the AWS SDK does this for you.

```
$ curl http://169.254.169.254/latest/meta-data/iam/security-creden
tials/role_name
```

Launching an Instance with an IAM Role Using an AWS SDK

If you use an AWS SDK to write your application, you automatically get temporary security credentials from the role associated with the current instance. The AWS SDK documentation includes walkthroughs that show how an application can use security credentials from a IAM role to read an Amazon S3 bucket. For more information, see the following topics in the SDK documentation:

- Using IAM Roles for EC2 Instances with the SDK for Java
- Using IAM Roles for EC2 Instances with the SDK for PHP
- Using IAM Roles for EC2 Instances with the SDK for Ruby

Authorizing Inbound Traffic for Your Instances

Security groups enable you to control traffic to your instance, including the kind of traffic that can reach your instance. For example, you can allow computers from only your home network to access your instance via SSH. If your instance is a web server, you can allow all IP addresses to access your instance via HTTP, so that external users can browse the content on your web server.

To enable network access to your instance, you must allow inbound traffic to your instance. To open a port for inbound traffic, add a rule to a security group that you associated with your instance when you launched it.

To connect to your instance, you must set up a rule to authorize SSH traffic from your computer's public IP address. To allow SSH traffic from additional IP address ranges, add another rule for each range you need to authorize.

Before You Start

Decide who requires access to your instance; for example, a single host or a specific network that you trust. In this case, we use your local system's public IP address. You can get the public IP address of your local computer using a service. For example, we provide the following service: http://checkip.amazon-aws.com. To locate another service that provides your IP address, use the search phrase "what is my IP address". If you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

> **Caution**
> If you use `0.0.0.0/0`, you enable all IP addresses to access your instance using SSH. This is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, you'll authorize only a specific IP address or range of addresses to access your instance.

For more information about security groups, see Amazon EC2 Security Groups (p. 421).

Adding a Rule for Inbound SSH Traffic to a Linux Instance

Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level. You must add rules to a security group that enable you to connect to your Linux instance from your IP address using SSH.

To add a rule to a security group for inbound SSH traffic using the console

1. In the navigation pane of the Amazon EC2 console, click **Instances**. Select your instance and look at the **Description** tab; **Security groups** lists the security groups that are associated with the instance. Click **view rules** to display a list of the rules that are in effect for the instance.
2. In the navigation pane, click **Security Groups**. Select one of the security groups associated with your instance.
3. In the details pane, on the **Inbound** tab, click **Edit**. In the dialog, click **Add Rule**, and then select **SSH** from the **Type** list.
4. In the **Source** field, specify the public IP address of your computer, in CIDR notation. For example, if your IP address is `203.0.113.25`, specify `203.0.113.25/32` to list this single IP address in CIDR notation. If your company allocates addresses from a range, specify the entire range, such as `203.0.113.0/24`.

   For information about finding your IP address, see Before You Start (p. 467).
5. Click **Save**.

To add a rule to a security group using the command line

You can use one of the following commands. Be sure to run this command on your local system, not on the instance itself. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- authorize-security-group-ingress (AWS CLI)
- ec2-authorize (Amazon EC2 CLI)
- Grant-EC2SecurityGroupIngress (AWS Tools for Windows PowerShell)

Assigning a Security Group to an Instance

You can assign a security group to an instance when you launch the instance. When you add or remove rules, those changes are automatically applied to all instances to which you've assigned the security group.

After you launch an instance in EC2-Classic, you can't change its security groups. After you launch an instance in a VPC, you can change its security groups. For more information, see Changing an Instance's Security Groups in the *Amazon VPC User Guide*.

Amazon EC2 and Amazon Virtual Private Cloud (VPC)

Amazon Virtual Private Cloud (Amazon VPC) enables you to define a virtual network in your own logically isolated area within the 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 data center, with the benefits of using AWS's scalable infrastructure. You can configure your VPC; you can select its IP address range, create subnets, and configure route tables, network gateways, and security settings. You can connect instances in your VPC to the Internet. You can connect your VPC to your own corporate data center, making the AWS cloud an extension of your data center. To protect the resources in each subnet, you can use multiple layers of security, including security groups and network access control lists. For more information, see the Amazon VPC 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 (p. 471).

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 (from Amazon's public IP address pool)	Your instance receives a public IP address.	Your instance launched in a default subnet receives a public IP address by default, unless you specify otherwise during launch, or you modify the subnet's public IP address attribute.	Your instance doesn't receive a public IP address by default, unless you specify otherwise during launch, or you modify the subnet's public IP address attribute.
Private IP address	Your instance receives a private IP address from the EC2-Classic range each time it's started.	Your instance receives a static private IP address from the address range of your default VPC.	Your instance receives a static private IP address from the address range of your VPC.
Multiple private IP addresses	We select a single private IP address for your instance; multiple IP addresses are not supported.	You can assign multiple private IP addresses to your instance.	You can assign multiple private IP addresses to your instance.
Elastic IP address	An EIP is disassociated from your instance when you stop it.	An EIP remains associated with your instance when you stop it.	An EIP remains associated with your instance when you stop it.
DNS hostnames	DNS hostnames are enabled by default.	DNS hostnames are enabled by default.	DNS hostnames are disabled by default.
Security group	A security group can reference security groups that belong to other AWS accounts. You can create up to 500 security groups in each region.	A security group can reference security groups for your VPC only. You can create up to 100 security groups per VPC.	A security group can reference security groups for your VPC only. You can create up to 100 security groups per VPC.

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
Security group association	You can assign an unlimited number of security groups to an instance when you launch it. You can't change the security groups of your running instance. You can either modify the rules of the assigned security groups, or replace the instance with a new one (create an AMI from the instance, launch a new instance from this AMI with the security groups that you need, disassociate any Elastic IP address from the original instance and associate it with the new instance, and then terminate the original instance).	You can assign up to 5 security groups to an instance. You can assign security groups to your instance when you launch it and while it's running.	You can assign up to 5 security groups to an instance. You can assign security groups to your instance when you launch it and while it's running.
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.
Tenancy	Your instance runs on shared hardware.	You can run your instance on shared hardware or single-tenant hardware.	You can run your instance on shared hardware or single-tenant hardware.

Amazon VPC Documentation

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

Guide	Description
Amazon VPC Getting Started Guide	Provides a hands-on introduction to Amazon VPC.
Amazon VPC User Guide	Provides detailed information about how to use Amazon VPC.
Amazon VPC Network Administrator Guide	Helps network administrators configure your customer gateway.

Supported Platforms

Amazon EC2 supports the following platforms. Your AWS account is capable of launching instances either into both platforms or only into EC2-VPC, on a region by region basis.

Platform	Introduced In	Description
EC2-Classic	The original release of Amazon EC2	Your instances run in a single, flat network that you share with other customers.
EC2-VPC	The original release of Amazon VPC	Your instances run in a virtual private cloud (VPC) that's logically isolated to your AWS account.

For more information about the availability of either platform in your account, see Availability in the *Amazon VPC User Guide*.

Supported Platforms in the Amazon EC2 Console

The Amazon EC2 console indicates which platforms you can launch instances into for the selected region, and whether you have a default VPC in that region.

Verify that the region you'll use is selected in the navigation bar. On the Amazon EC2 console dashboard, look for **Supported Platforms** under **Account Attributes**. If there are two values, EC2 and VPC, you can launch instances into either platform. If there is one value, VPC, you can launch instances only into EC2-VPC.

If you can launch instances only into EC2-VPC, we create a default VPC for you. Then, when you launch an instance, we launch it into your default VPC, unless you create a nondefault VPC and specify it when you launch the instance.

EC2-VPC

The dashboard displays the following under **Account Attributes** to indicate that the account supports only the EC2-VPC platform, and has a default VPC with the identifier vpc-1a2b3c4d.

Supported Platforms
  VPC

Default VPC
  vpc-1a2b3c4d

If your account supports only EC2-VPC, you can select a VPC from the **Network** list, and a subnet from the **Subnet** list when you launch an instance using the launch wizard.

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

EC2-Classic, EC2-VPC

The dashboard displays the following under **Account Attributes** to indicate that the account supports both the EC2-Classic and EC2-VPC platforms.

Supported Platforms
  EC2
  VPC

If your account supports EC2-Classic and EC2-VPC, you can launch into EC2-Classic using the launch wizard by selecting **Launch into EC2-Classic** from the **Network** list. To launch into a VPC, you can select a VPC from the **Network** list, and a subnet from the **Subnet** list.

Related Topic

For more information about how you can tell which platforms you can launch instances into, see Detecting Your Supported Platforms in the *Amazon VPC User Guide*.

Differences Between Instances in EC2-Classic and EC2-VPC

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

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

The following diagram shows instances in each platform. Note the following:

- Instances C1, C2, C3, and C4 are in the EC2-Classic platform. C1 and C2 were launched by one account, and C3 and C4 were launched by a different account. These instances can communicate with each other, can access the Internet directly, and can access other services such as Amazon Simple Storage Service (Amazon S3).
- Instances V1 and V2 are in different subnets in the same VPC in the EC2-VPC platform. They were launched by the account that owns the VPC; no other account can launch instances in this VPC. These instances can communicate with each other and can access the following through the Internet gateway: instances in EC2-Classic, other services (such as Amazon S3), and the Internet.

For more information about the differences between EC2-Classic and EC2-VPC, see Amazon EC2 and Amazon Virtual Private Cloud (VPC) (p. 468).

Migrating from EC2-Classic to a VPC
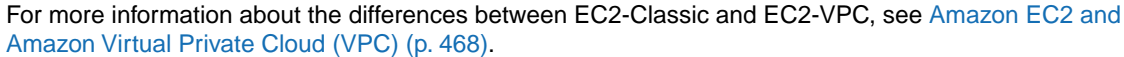
Your AWS account might support both EC2-Classic and EC2-VPC, depending on when you created your account and which regions you've used. For more information, and to find out which platform your account supports, see Supported Platforms (p. 471). For more information about the benefits of using a VPC, and the differences between EC2-Classic and EC2-VPC, see Amazon EC2 and Amazon Virtual Private Cloud (VPC) (p. 468).

You create and use resources in your AWS account. Some resources and features, such as enhanced networking and T2 instances, can be used only in a VPC. Some resources in your account can be shared

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

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

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

> **Note**
> If you use Reserved Instances, you can change the network platform for your Reserved Instances from EC2-Classic to EC2-VPC. For more information, see Modifying Your Reserved Instances (p. 218).

Migrating to a VPC

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

Topics

Step 1: Create a VPC

To start using a VPC, ensure that you have one in your account. You can create one using one of these methods:

- Use a new, EC2-VPC-only AWS account. Your EC2-VPC-only account comes with a default VPC in each region, which is ready for you to use. Instances that you launch are by default launched into this VPC, unless you specify otherwise. For more information about your default VPC, see Your Default VPC and Subnets. Use this option if you'd prefer not to set up a VPC yourself, or if you do not need specific requirements for your VPC configuration.
- In your existing AWS account, open the Amazon VPC console and use the VPC wizard to create a new VPC. For more information, see Scenarios for Amazon VPC. Use this option if you want to set up a VPC quickly in your existing EC2-Classic account, using one of the available configuration sets in the wizard. You'll specify this VPC each time you launch an instance.
- In your existing AWS account, open the Amazon VPC console and set up the components of a VPC according to your requirements. For more information, see Your VPC and Subnets. Use this option if you have specific requirements for your VPC, such as a particular number of subnets. You'll specify this VPC each time you launch an instance.

> **Note**
> T2 instance types must be launched into a VPC. If you do not have any VPCs in your EC2-Classic account, and you use the launch wizard in the Amazon EC2 console to launch a T2 instance, the wizard creates a nondefault VPC for you. For more information about T2 instance types, see T2 Instances (p. 105). Your T2 instance will not be able to communicate with your

EC2-Classic instances using private IP addresses. Consider migrating your existing instances to the same VPC using the methods outlined in this topic.

Step 2: Configure Your Security Group

You cannot use the same security groups between EC2-Classic and a VPC. However, if you want your instances in your VPC to have the same security group rules as your EC2-Classic instances, you can use the Amazon EC2 console to copy your existing EC2-Classic security group rules to a new VPC security group.

> **Important**
> You can only copy security group rules to a new security group in the same AWS account in the same region. If you've created a new AWS account, you cannot use this method to copy your existing security group rules to your new account. You'll have to create a new security group, and add the rules yourself. For more information about creating a new security group, see Amazon EC2 Security Groups (p. 421).

To copy your security group rules to a new security group

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Security Groups**.
3. Select the security group that's associated with your EC2-Classic instance, then click **Actions** and select **Copy to new**.
4. In the **Create Security Group** dialog box, specify a name and description for your new security group. Select your VPC from the **VPC** list.
5. The **Inbound** tab is populated with the rules from your EC2-Classic security group. You can modify the rules as required. In the **Outbound** tab, a rule that allows all outbound traffic has automatically been created for you. For more information about modifying security group rules, see Amazon EC2 Security Groups (p. 421).

    > **Note**
    > If you've defined a rule in your EC2-Classic security group that references another security group, you will not be able to use the same rule in your VPC security group. Modify the rule to reference a security group in the same VPC.

6. Click **Create**.

Step 3: Create an AMI from Your EC2-Classic Instance

An AMI is a template for launching your instance. You can create your own AMI based on an existing EC2-Classic instance, then use that AMI to launch instances into your VPC.

The method you use to create your AMI depends on the root device type of your instance, and the operating system platform on which your instance runs. To find out the root device type of your instance, go to the **Instances** page, select your instance, and look at the information in the **Root device type** field in the **Description** tab. If the value is `ebs`, then your instance is EBS-backed. If the value is `instance-store`, then your instance is instance store-backed. You can also use the describe-instances AWS CLI command to find out the root device type.

The following table provides options for you to create your AMI based on the root device type of your instance, and the software platform.

> **Important**
> Some instance types support both PV and HVM virtualization, while others support only one or the other. If you plan to use your AMI to launch a different instance type than your current instance type, check that the instance type supports the type of virtualization that your AMI offers. If your AMI supports PV virtualization, and you want to use an instance type that supports HVM virtual-

ization, you may have to reinstall your software on a base HVM AMI. For more information about PV and HVM virtualization, see Virtualization Types (p. 58).

Instance Root Device Type	Action
EBS	Create an EBS-backed AMI from your instance. For more information, see Creating an Amazon EBS-Backed Linux AMI (p. 74).
Instance store	Create an instance store-backed AMI from your instance using the AMI tools. For more information, see Creating an Instance Store-Backed Linux AMI (p. 77).
Instance store	Transfer your instance data to an EBS volume, then take a snapshot of the volume, and create an AMI from the snapshot. For more information, see Converting your Instance Store-Backed AMI to an Amazon EBS-Backed AMI (p. 82). **Note** This method converts an instance store-backed instance to an EBS-backed instance.

(Optional) Store Your Data on Amazon EBS Volumes

You can create an Amazon EBS volume and use it to back up and store the data on your instance—like you would use a physical hard drive. Amazon EBS volumes can be attached and detached from any instance in the same Availability Zone. You can detach a volume from your instance in EC2-Classic, and attach it to a new instance that you launch into your VPC in the same Availability Zone.

For more information about Amazon EBS volumes, see the following topics:

- Amazon EBS Volumes (p. 521)
- Creating an Amazon EBS Volume (p. 526)
- Attaching an Amazon EBS Volume to an Instance (p. 529)

To back up the data on your Amazon EBS volume, you can take periodic snapshots of your volume. If you need to, you can restore an Amazon EBS volume from your snapshot. For more information about Amazon EBS snapshots, see the following topics:

- Amazon EBS Snapshots (p. 559)
- Creating an Amazon EBS Snapshot (p. 560)
- Restoring an Amazon EBS Volume from a Snapshot (p. 527)

Step 4: Launch an Instance Into Your VPC

After you've created an AMI, you can launch an instance into your VPC. The instance will have the same data and configurations as your existing EC2-Classic instance.

You can either launch your instance into a VPC that you've created in your existing account, or into a new, VPC-only AWS account.

Using Your Existing EC2-Classic Account

You can use the Amazon EC2 launch wizard to launch an instance into your VPC.

To launch an instance into your VPC

1. Open the Amazon EC2 console.
2. On the dashboard, click **Launch Instance**.
3. On the **Choose an Amazon Machine Image** page, select the **My AMIs** category, and select the AMI you created.
4. On the **Choose an Instance Type** page, select the type of instance, and click **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, select your VPC from the **Network** list. Select the required subnet from the **Subnet** list. Configure any other details you require, then click through the next pages of the wizard until you reach the **Configure Security Group** page.
6. Select **Select an existing group**, and select the security group you created earlier. Click **Review and Launch**.
7. Review your instance details, then click **Launch** to specify a key pair and launch your instance.

For more information about the parameters you can configure in each step of the wizard, see Launching an Instance (p. 290).

Using Your New, VPC-Only Account

To launch an instance in your new AWS account, you'll first have to share the AMI you created with your new account. You can then use the Amazon EC2 launch wizard to launch an instance into your default VPC.

To share an AMI with your new AWS account

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

To launch an instance into your default VPC

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

For more information about the parameters you can configure in each step of the wizard, see Launching an Instance (p. 290).

Example: Migrating a Simple Web Application

In this example, you use AWS to host your gardening website. To manage your website, you have three running instances in EC2-Classic. Instances A and B host your public-facing web application, and you use an Elastic Load Balancer to load balance the traffic between these instances. You've assigned Elastic IP addresses to instances A and B so that you have static IP addresses for configuration and administration tasks on those instances. Instance C holds your MySQL database for your website. You've registered the domain name `www.garden.example.com`, and you've used Amazon Route 53 to create a hosted zone with an alias record set that's associated with the DNS name of your load balancer.
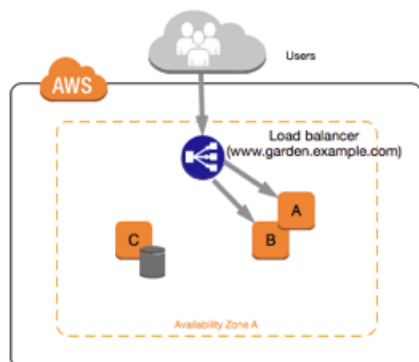
The first part of migrating to a VPC is deciding what kind of VPC architecture will suit your needs. In this case, you've decided on the following: one public subnet for your web servers, and one private subnet for your database server. As your website grows, you can add more web servers and database servers to your subnets. By default, instances in the private subnet cannot access the Internet; however, you can enable Internet access through a Network Address Translation (NAT) instance in the public subnet. You may want to set up a NAT instance to support periodic updates and patches from the Internet for your database server. You'll assign new Elastic IP addresses to your web servers, and create an Elastic Load Balancer in your public subnet to load balance the traffic between your web servers.
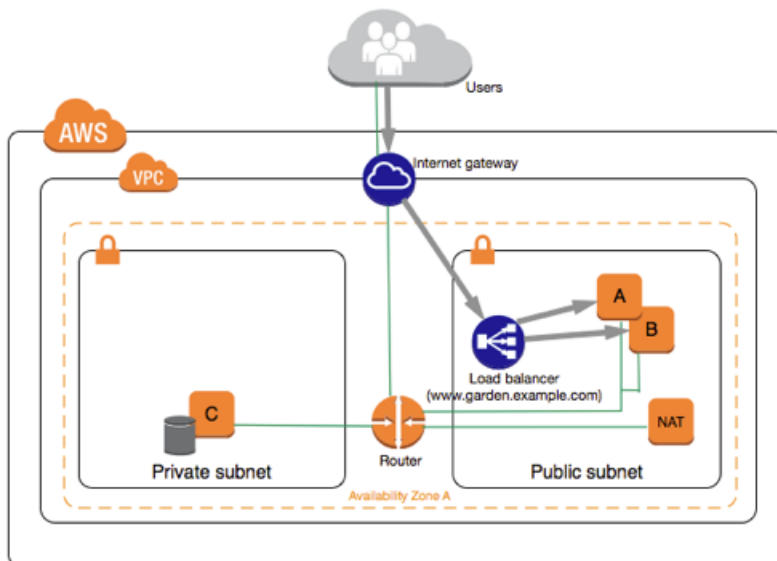
To migrate your web application to a VPC, you can follow these steps:

- **Create a VPC**: In this case, you can use the VPC wizard in the Amazon VPC console to create your VPC and subnets. The second wizard configuration creates a VPC with one private and one public

subnet, and launches and configures a NAT instance in your public subnet for you. For more information, see Scenario 2: VPC with Public and Private Subnets in the *Amazon VPC User Guide*.

- **Create AMIs from your instances**: Create an AMI from one of your web servers, and a second AMI from your database server. For more information, see Step 3: Create an AMI from Your EC2-Classic Instance (p. 475).

- **Configure your security groups**: In your EC2-Classic environment, you have one security group for your web servers, and another security group for your database server. You can use the Amazon EC2 console to copy the rules from each security group into new security groups for your VPC. For more information, see Step 2: Configure Your Security Group (p. 475).

   Tip
   Create the security groups that are referenced by other security groups first.

- **Launch an instance into your new VPC**: Launch replacement web servers into your public subnet, and launch your replacement database server into your private subnet. For more information, see Step 4: Launch an Instance Into Your VPC (p. 476).

- **Create new Elastic IP addresses**: You cannot use your EC2-Classic Elastic IP addresses in a VPC. Instead, create new Elastic IP addresses for use in a VPC, and assign them to your web servers. For more information, see Elastic IP Addresses (EIP) (p. 489).

- **Configure your NAT instance**: If you want to make use of your NAT instance to allow your database server to access the Internet, you'll have to create a security group for your NAT instance that allows HTTP and HTTPS traffic from your private subnet. For more information, see NAT Instances.

- **Configure your database**: When you created an AMI from your database server in EC2-Classic, all the configuration information that was stored in that instance was copied to the AMI. You may have to connect to your new database server and update the configuration details; for example, if you configured your database to grant full read, write, and modification permissions to your web servers in EC2-Classic, you'll have to update the configuration files to grant the same permissions to your new VPC web servers instead.

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

- **Configure your web servers**: Your web servers will have the same configuration settings as your instances in EC2-Classic. For example, if you configured your web servers to use the database in EC2-Classic, update your web servers' configuration settings to point to your new database instance.

   Note
   By default, instances launched into a nondefault subnet are not assigned a public IP address, unless you specify otherwise at launch. Your new database server may not have a public IP address. In this case, you can update your web servers' configuration file to use your new database server's private DNS name. Instances in the same VPC can communicate with each other via private IP address.

- **Create a new load balancer**: To continue using Elastic Load Balancing to load balance the traffic to your instances, make sure you understand the various ways you can configure your load balancer in VPC. For more information, see Elastic Load Balancing in Amazon VPC.

- **Update your DNS records**: After you've set up your load balancer in your public subnet, ensure that your `www.garden.example.com` domain points to your new load balancer. To do this, you'll need to update your DNS records and update your alias record set in Amazon Route 53. For more information about using Amazon Route 53, see Getting Started with Amazon Route 53.

- **Shut down your EC2-Classic resources**: After you've verified that your web application is working from within the VPC architecture, you can shut down your EC2-Classic resources to stop incurring charges for them. Terminate your EC2-Classic instances, and release your EC2-Classic Elastic IP addresses.

Amazon EC2 Instance IP Addressing

We provide your instances with IP addresses and DNS hostnames. These can vary depending on whether you launched the instance in the EC2-Classic platform or in a virtual private cloud (VPC).

For information about the EC2-Classic and EC2-VPC platforms, see Supported Platforms (p. 471). For information about Amazon VPC, see What is Amazon VPC? in the *Amazon VPC User Guide*.

Contents

- Private IP Addresses and Internal DNS Hostnames (p. 480)
- Public IP Addresses and External DNS Hostnames (p. 481)
- Elastic IP Addresses (p. 482)
- Differences Between EC2-Classic and EC2-VPC (p. 482)
- Determining Your Public, Private, and Elastic IP Addresses (p. 482)
- Assigning a Public IP Address (p. 484)
- Multiple Private IP Addresses (p. 485)

Private IP Addresses and Internal DNS Hostnames

A private IP address is an IP address that's not reachable over the Internet. You can use private IP addresses for communication between instances in the same network (EC2-Classic or a VPC). For more information about the standards and specifications of private IP addresses, go to RFC 1918.

When you launch an instance, we allocate a private IP address for the instance using DHCP. Each instance is also given an internal DNS hostname that resolves to the private IP address of the instance; for example, `ip-10-251-50-12.ec2.internal`. You can use the internal DNS hostname for communication between instances in the same network, but we can't resolve the DNS hostname outside the network that the instance is in.

An instance launched in a VPC is given a primary private IP address in the address range of the subnet. For more information, see Subnet Sizing in the *Amazon VPC User Guide*. If you don't specify a primary private IP address when you launch the instance, we select an available IP address in the subnet's range for you. Each instance in a VPC has a default network interface (eth0) that is assigned the primary private IP address. You can also specify additional private IP addresses, known as *secondary private IP addresses*. Unlike primary private IP addresses, secondary private IP addresses can be reassigned from one instance to another. For more information, see Multiple Private IP Addresses (p. 485).

For instances launched in EC2-Classic, we release the private IP address when the instance is stopped or terminated. If you restart your stopped instance, it receives a new private IP address.

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.

If you create a custom firewall configuration in EC2-Classic, you must create a rule in your security group that allows inbound traffic from port 53 (DNS)—with a destination port from the ephemeral range—from the address of the Amazon DNS server; otherwise, internal DNS resolution from your instances fails. If your firewall doesn't automatically allow DNS query responses, then you'll need to allow traffic from the IP address of the Amazon DNS server. To get the IP address of the Amazon DNS server, use the following command from within your instance:

```
grep nameserver /etc/resolv.conf
```

Public IP Addresses and External DNS Hostnames

A public IP address is reachable from the Internet. You can use public IP addresses for communication between your instances and the Internet or other AWS products, such as Amazon Simple Storage Service (Amazon S3).

Each instance that receives a public IP address is also given an external DNS hostname; for example, `ec2-203-0-113-25.compute-1.amazonaws.com`. 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. The public IP address is mapped to the primary private IP address through network address translation (NAT). For more information about NAT, go to RFC 1631: The IP Network Address Translator (NAT).

When you launch an instance in EC2-Classic, we automatically assign a public IP address to the instance. You cannot modify this behavior. When you launch an instance into a VPC, your subnet has an attribute that determines whether instances launched into that subnet receive a public IP address. By default, we don't automatically assign a public IP address to an instance that you launch in a nondefault subnet.

You can control whether your instance in a VPC receives a public IP address by doing the following:

*   Modifying the public IP addressing attribute of your subnet. For more information, see Modifying Your Subnet's Public IP Addressing Behavior in the *Amazon VPC User Guide*.
*   Enabling or disabling the public IP addressing feature during launch, which overrides the subnet's public IP addressing attribute. For more information, see Assigning a Public IP Address (p. 484).

A public IP address is assigned to your instance from Amazon's pool of public IP addresses, and is not associated with your AWS account. When a public IP address is disassociated from your instance, it is released back into the public IP address pool, and you cannot reuse it.

You cannot manually associate or disassociate a public IP address from your instance. Instead, in certain cases, we release the public IP address from your instance, or assign it a new one:

*   We release the public IP address for your instance when it's stopped or terminated. Your stopped instance receives a new public IP address when it's restarted.
*   We release the public IP address for your instance when you associate an Elastic IP address with your instance, or when you associate an Elastic IP address with the primary network interface (eth0) of your instance in a VPC. When you disassociate the Elastic IP address from your instance, it receives a new public IP address.
*   If the public IP address of your instance in a VPC has been released, it will not receive a new one if there is more than one network interface attached to your instance.

If you require a persistent public IP address that can be associated to and from instances as you require, use an Elastic IP address instead. You can allocate your own Elastic IP address, and associate it with your instance. For more information, see Elastic IP Addresses (EIP) (p. 489).

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 VPC User Guide*.

> **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.

Elastic IP Addresses

An Elastic IP address is a public IP address that you can allocate to your account. You can associate it to and from instances as you require, and it's allocated to your account until you choose to release it. For more information about Elastic IP addresses and how to use them, see Elastic IP Addresses (EIP) (p. 489).

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 Subnet	Nondefault Subnet
Public IP address (from Amazon's public IP address pool)	Your instance receives a public IP address.	Your instance receives a public IP address by default, unless you specify otherwise during launch, or you modify the subnet's public IP address attribute.	Your instance doesn't receive a public IP address by default, unless you specify otherwise during launch, or you modify the subnet's public IP address attribute.
Private IP address	Your instance receives a private IP address from the EC2-Classic range each time it's started.	Your instance receives a static private IP address from the address range of your default subnet.	Your instance receives a static private IP address from the address range of your subnet.
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 Elastic IP address is disassociated from your instance when you stop it.	An Elastic IP address remains associated with your instance when you stop it.	An Elastic IP address 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, except if you've created your VPC using the VPC wizard in the Amazon VPC console.

Determining Your Public, Private, and Elastic IP Addresses

You can use the Amazon EC2 console to determine the private IP addresses, public IP addresses, and Elastic IP addresses of your instances. You can also determine the public and private IP addresses of your instance from within your instance by using instance metadata. For more information, see Instance Metadata and User Data (p. 256).

To determine your instance's private IP addresses using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**.
3. Select your instance. In the details pane, get the private IP address from the **Private IPs** field, and get the internal DNS hostname from the **Private DNS** field.
4. (VPC only) If you have one or more secondary private IP addresses assigned to network interfaces that are attached to your instance, get those IP addresses from the **Secondary private IPs** field.
5. (VPC only) Alternatively, in the navigation pane, click **Network Interfaces**, and then select the a network interface that's associated with your instance.
6. Get the primary private IP address from the **Primary private IP** field, and the internal DNS hostname from the **Private DNS** field.
7. If you've assigned secondary private IP addresses to the network interface, get those IP addresses from the **Secondary private IPs** field.

To determine your instance's public IP addresses using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**.
3. Select your instance. In the details pane, get the public IP address from the **Public IP** field, and get the external DNS hostname from the **Public DNS** field.

   Note
   If you've associated an Elastic IP address with your instance, the **Public IP** field displays the Elastic IP address.

4. If an Elastic IP address has been associated with the instance, get the Elastic IP address from the **Elastic IP** field.
5. (VPC only) Alternatively, in the navigation pane, click **Network Interfaces**, and then select a network interface that's associated with your instance.
6. Get the public IP address from the **Public IPs** field. An asterisk (*) indicates the public IP address or Elastic IP address that's mapped to the primary private IP address.

   Note
   The public IP address is displayed as a property of the network interface in the console, but it's mapped to the primary private IP address through NAT. Therefore, if you inspect the properties of your network interface on your instance, for example, through `ifconfig`, the public IP address is not displayed. To determine your instance's public IP address from within the instance, you can use instance metadata.

To determine your instance's IP addresses using instance metadata

1. Connect to your instance.
2. Use the following command to access the private IP address:

```
$ curl http://169.254.169.254/latest/meta-data/local-ipv4
```

3. Use the following command to access the public IP address:

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

Note that if an Elastic IP address is associated with the instance, the value returned is that of the Elastic IP address.

Assigning a Public IP Address

If you launch an instance in EC2-Classic, it is assigned a public IP address by default. You can't modify this behavior.

In a VPC, all subnets have an attribute that determines whether instances launched into that subnet are assigned a public IP address. By default, nondefault subnets have this attribute set to false, and default subnets have this attribute set to true. If you launch an instance into a VPC, a public IP addressing feature is available for you to control whether your instance is assigned a public IP address - you can override the default behavior of the subnet's IP addressing attribute. The public IP address is assigned from Amazon's pool of public IP addresses, and is assigned to the network interface with the device index of eth0. This feature depends on certain conditions at the time you launch your instance.

> **Important**
> You can't manually disassociate the public IP address from your instance after launch. Instead, it's automatically released in certain cases, after which you cannot reuse it. For more information, see Public IP Addresses and External DNS Hostnames (p. 481). If you require a persistent public IP address that you can associate or disassociate at will, assign an Elastic IP address to the instance after launch instead. For more information, see Elastic IP Addresses (EIP) (p. 489).

To access the public IP addressing feature when launching an instance

1. Open the Amazon EC2 console.
2. Click **Launch Instance**.
3. Choose an AMI and click its **Select** button, then choose an instance type and click **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, select a VPC from the **Network** list. An **Auto-assign Public IP** list is displayed. Select **Enable** or **Disable** to override the default setting for the subnet.

   > **Important**
   > A public IP address can only be assigned to a single, new network interface with the device index of eth0. The **Auto-assign Public IP** list is not available if you're launching with multiple network interfaces, or if you select an existing network interface for eth0.

5. Follow the steps on the next pages of the wizard to complete your instance's setup. For more information about the wizard configuration options, see Launching an Instance (p. 290). On the final **Review Instance Launch** page, review your settings, and then click **Launch** to choose a key pair and launch your instance.
6. On the **Instances** page, select your new instance and view its public IP address in **Public IP** field in the details pane.

The public IP addressing feature is only available during launch. However, whether you assign a public IP address to your instance during launch or not, you can associate an Elastic IP address with your instance after it's launched. For more information, see Elastic IP Addresses (EIP) (p. 489). You can also modify your subnet's public IP addressing behavior. For more information, see Modifying Your Subnet's Public IP Addressing Behavior.

API and Command Line Tools for Public IP Addressing

To enable or disable the public IP addressing feature, use one of the methods in the table below. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

Method	Parameter
AWS CLI	Use the `--associate-public-ip-address` or the `--no-associate-public-ip-address` option with the run-instances command.
Amazon EC2 CLI	Use the `--associate-public-ip-address` option with the ec2-run-instances command.
AWS Tools for Windows PowerShell	Use the `-AssociatePublicIp` parameter with the New-EC2Instance command.
Query API	Use the `NetworkInterface.n.AssociatePublicIpAddress` parameter with the RunInstances request.

Multiple Private IP Addresses

In EC2-VPC, you can specify multiple private IP addresses for your instances. The number of network interfaces and private IP addresses that you can specify for an instance depends on the instance type. For more information, see Private IP Addresses Per ENI Per Instance Type (p. 495).

It can be useful to assign multiple private IP addresses to an instance in your VPC to do the following:

- Host multiple websites on a single server by using multiple SSL certificates on a single server and associating each certificate with a specific IP address.
- Operate network appliances, such as firewalls or load balancers, that have multiple private IP addresses for each network interface.
- Redirect internal traffic to a standby instance in case your instance fails, by reassigning the secondary private IP address to the standby instance.

Contents

How Multiple IP Addresses Work

The following list explains how multiple IP addresses work with network interfaces:

- You can assign a secondary private IP address to any network interface. The network interface can be attached to or detached from the instance.
- You must choose a secondary private IP address that's in the CIDR block range of the subnet for the network interface.
- Security groups apply to network interfaces, not to IP addresses. Therefore, IP addresses are subject to the security group of the network interface in which they're specified.
- Secondary private IP addresses can be assigned and unassigned to elastic network interfaces attached to running or stopped instances.

- Secondary private IP addresses that are assigned to a network interface can be reassigned to another one if you explicitly allow it.
- When assigning multiple secondary private IP addresses to a network interface using the command line tools or API, the entire operation fails if one of the secondary private IP addresses can't be assigned.
- Primary private IP addresses, secondary private IP addresses, and any associated Elastic IP addresses remain with the network interface when it is detached from an instance or attached to another instance.
- Although you can't move the primary network interface from an instance, you can reassign the secondary private IP address of the primary network interface to another network interface.
- You can move any additional network interface from one instance to another.

The following list explains how multiple IP addresses work with Elastic IP addresses:

- Each private IP address can be associated with a single Elastic IP address, and vice versa.
- When a secondary private IP address is reassigned to another interface, the secondary private IP address retains its association with an Elastic IP address.
- When a secondary private IP address is unassigned from an interface, an associated Elastic IP address is automatically disassociated from the secondary private IP address.

Assigning a Secondary Private IP Address

You can assign the secondary private IP address to the network interface for an instance as you launch the instance, or after the instance is running.

To assign a secondary private IP address when launching an instance in EC2-VPC

1. Open the Amazon EC2 console.
2. Click the **Launch Instance** button.
3. Choose an AMI and click its **Select** button, then choose an instance type and click **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, choose a VPC from the **Network** list, and a subnet from the **Subnet** list.
5. In the **Network Interfaces** section, do the following, and then click **Next: Add Storage**:

    a. Click **Add Device** to add another network interface. The console enables you specify up to 2 network interfaces when you launch an instance. After you launch the instance, click **Network Interfaces** in the navigation pane to add additional network interfaces. The total number of network interfaces that you can attach varies by instance type. For more information, see Private IP Addresses Per ENI Per Instance Type (p. 495).

    b. For each network interface, you can specify a primary private IP address, and one or more secondary private IP addresses. For this example, however, accept the IP address that we automatically assign.

    c. Under **Secondary IP addresses**, click **Add IP**, and then enter a private IP address in the subnet range, or accept the default, `Auto-assign`, to let us select an address.

        Important
        After you have added a secondary private IP address to a network interface, you must connect to the instance and configure the secondary private IP address on the instance itself. For more information, see Configuring the Operating System on Your Instance to Recognize the Secondary Private IP Address (p. 487).

6. On the next **Add Storage** page, you can specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume), and then click **Next: Tag Instance**.

7. On the **Tag Instance** page, specify tags for the instance, such as a user-friendly name, and then click **Next: Configure Security Group**.

8. On the **Configure Security Group** page, select an existing security group or create a new one. Click **Review and Launch**.

9. On the **Review Instance Launch** page, review your settings, and then click **Launch** to choose a key pair and launch your instance. If you're new to Amazon EC2 and haven't created any key pairs, the wizard prompts you to create one.

To assign a secondary IP address during launch using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- The `--secondary-private-ip-addresses` option with the run-instances command (AWS CLI)
- The `--secondary-private-ip-address` option with the ec2-run-instances command (Amazon EC2 CLI)

To assign a secondary private IP to an existing instance

1. Open the Amazon EC2 console.

2. In the navigation pane, click **Network Interfaces**, and then right-click the network interface attached to the instance.

3. Select **Manage Private IP Addresses**.

4. In the **Manage Private IP Addresses** dialog box, do the following:

    a. Click **Assign new IP**.

    b. Enter a specific IP address that's within the subnet range for the instance, or leave the field blank and we'll select an IP address for you.

    c. (Optional) Select **Allow reassignment** to allow the secondary private IP address to be reassigned if it is already assigned to another network interface.

    d. Click **Yes, Update**, and then click **Close**.

    > **Note**
    > You can also assign a secondary private IP address to an instance by clicking **Instances** in the navigation pane, right-clicking your instance, and selecting **Manage Private IP Addresses**. You can configure the same information in the dialog as you did in the steps above.

To assign a secondary private IP to an existing instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- assign-private-ip-addresses (AWS CLI)
- ec2-assign-private-ip-addresses (Amazon EC2 CLI)
- Register-EC2PrivateIpAddress (AWS Tools for Windows PowerShell)

Configuring the Operating System on Your Instance to Recognize the Secondary Private IP Address

After you assign a secondary private IP address to your instance, you need to configure the operating system on your instance to recognize the secondary private IP address.

If you are using Amazon Linux, the ec2-net-utils package can take care of this step for you. It configures additional network interfaces that you attach 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 your network configuration, you can remove the ec2-net-utils package. For more information, see Configuring Your Network Interface Using ec2-net-utils (p. 498).

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.

Associating an Elastic IP Address with the Secondary Private IP Address

To associate an Elastic IP address with a secondary private IP address in EC2-VPC

1. Open the Amazon EC2 console.
2. Click **Elastic IPs** in the navigation pane.
3. Right-click the IP address, and then click **Associate**.
4. In the **Associate Address** dialog box, select the network interface from the **Network Interface** drop-down list, and then select the secondary IP address from the **Private IP address** drop-down list.
5. Click **Associate**.

To associate an Elastic IP address with a secondary private IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

*   associate-address (AWS CLI)
*   ec2-associate-address (Amazon EC2 CLI)
*   Register-EC2Address (AWS Tools for Windows PowerShell)

Viewing Your Secondary Private IP Addresses

To view the private IP addresses assigned to a network interface in EC2-VPC

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Select the network interface whose private IP addresses you want to view.
4. On the **Details** tab in the details pane, check the **Primary private IP** and **Secondary private IPs** fields for the primary private IP address and any secondary private IP addresses assigned to the network interface.

To view the private IP addresses assigned to an instance

1. Open the Amazon EC2 console.
2. Click **Instances** in the navigation pane.
3. Select the instance whose private IP addresses you want to view.
4. On the **Description** tab in the details pane, check the **Private IPs** and **Secondary private IPs** fields for the primary private IP address and any secondary private IP addresses assigned to the instance through its network interface.

Unassigning a Secondary Private IP Address

If you no longer require a secondary private IP address, you can unassign it from the instance or the network interface. When a secondary private IP address is unassigned from an elastic network interface, the Elastic IP address (if it exists) is also disassociated.

To unassign a secondary private IP address from an instance

1.   Open the Amazon EC2 console.
2.   Click **Instances** in the navigation pane.
3.   Right-click an instance, and then click **Manage Private IP Addresses**.
4.   In the **Manage Private IP Addresses** dialog box, beside the secondary private IP address to unassign, click **Unassign**.
5.   Click **Yes, Update**, and then close the dialog box.

To unassign a secondary private IP address from a network interface

1.   Open the Amazon EC2 console.
2.   Click **Network Interfaces** in the navigation pane.
3.   Right-click a network interface, and then click **Manage Private IP Addresses**.
4.   In the **Manage Private IP Addresses** dialog box, beside the secondary private IP address to unassign, click **Unassign**.
5.   Click **Yes, Update**, and then click **Close**.

To unassign a secondary private IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

*   unassign-private-ip-addresses (AWS CLI)
*   ec2-unassign-private-ip-addresses (Amazon EC2 CLI)
*   Unregister-EC2PrivateIpAddress (AWS Tools for Windows PowerShell)

Elastic IP Addresses (EIP)

An *Elastic IP address* (EIP) is a static IP address designed for dynamic cloud computing. With an EIP, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account. Your EIP is associated with your AWS account, not a particular instance, and it remains associated with your account until you choose to explicitly release it.

There's one pool of EIPs for use with the EC2-Classic platform and another for use with your VPC. You can't associate an EIP that you allocated for use with a VPC with an instance in EC2-Classic, and vice-versa. For more information about EC2-Classic and EC2-VPC, see Supported Platforms (p. 471).

Topics

Elastic IP Addresses in EC2-Classic

By default, we assign each instance in EC2-Classic two IP addresses at launch: a private IP address and a public IP address that is mapped to the private IP address through network address translation (NAT). The public IP address is allocated from the EC2-Classic public IP address pool, and is associated with your instance, not with your AWS account. You cannot reuse a public IP address after it's been disassociated from your instance.

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

When you associate an EIP with an instance, the instance's current public IP address is released to the EC2-Classic public IP address pool. If you disassociate an EIP from the instance, the instance is automatically assigned a new public IP address within a few minutes. In addition, stopping the instance also disassociates the EIP from it.

To ensure efficient use of EIPs, we impose a small hourly charge if an EIP is not associated with a running instance. For more information, see Amazon EC2 Pricing.

Elastic IP Addresses in a VPC

We assign each instance in a default VPC two IP addresses at launch: a private IP address and a public IP address that is mapped to the private IP address through network address translation (NAT). The public IP address is allocated from the EC2-VPC public IP address pool, and is associated with your instance, not with your AWS account. You cannot reuse a public IP address after it's been disassociated from your instance.

We assign each instance in a nondefault VPC only a private IP address, unless you specifically request a public IP address during launch, or you modify the subnet's public IP address attribute. To ensure that an instance in a nondefault VPC that has not been assigned a public IP address can communicate with the Internet, you must allocate an Elastic IP address for use with a VPC, and then associate that EIP with the elastic network interface (ENI) attached to the instance.

When you associate an EIP with an instance in a default VPC, or an instance in which you assigned a public IP to the eth0 network interface during launch, its current public IP address is released to the EC2-VPC public IP address pool. If you disassociate an EIP from the instance, the instance is automatically assigned a new public IP address within a few minutes. However, if you have attached a second network interface to the instance, the instance is not automatically assigned a new public IP address; you'll have to associate an EIP with it manually. The EIP remains associated with the instance when you stop it.

To ensure efficient use of EIPs, we impose a small hourly charge if an EIP is not associated with a running instance, or if it is associated with a stopped instance or an unattached network interface. While your instance is running, you are not charged for one EIP associated with the instance, but you are charged for any additional EIPs associated with the instance. For more information, see Amazon EC2 Pricing.

For information about using an EIP with an instance in a VPC, see Elastic IP Addresses in the *Amazon VPC 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 Elastic Network Interfaces (ENI) (p. 494).
Reassociation	If you try to associate an EIP that's already associated with another instance, the address is automatically associated with the new instance.	If your account supports EC2-VPC only, and you try to associate an EIP that's already associated with another instance, the address is automatically associated with the new instance. If you're using a VPC in an EC2-Classic account, and you try to associate an EIP that's already associated with another instance, it succeeds only if you allowed reassociation.
Instance stop	If you stop an instance, its EIP is disassociated, and you must re-associate the EIP when you restart the instance.	If you stop an instance, its EIP remains associated.
Multiple IP	Instances support only a single private IP address and a corresponding EIP.	Instances support multiple IP addresses, and each one can have a corresponding EIP. For more information, see Multiple Private IP Addresses (p. 485).

Allocating an Elastic IP Address

You can allocate an Elastic IP address using the AWS Management Console or the command line.

To allocate an Elastic IP address for use with EC2-Classic using the console

1. Open the Amazon EC2 console.
2. Click **Elastic IPs** in the navigation pane.
3. Click **Allocate New Address**.
4. Select `EC2` from the **EIP** list, and then click **Yes, Allocate**. Close the confirmation dialog box.

To allocate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- allocate-address (AWS CLI)
- ec2-allocate-address (Amazon EC2 CLI)

- New-EC2Address (AWS Tools for Windows PowerShell)

Describing Your Elastic IP Addresses

You can describe an Elastic IP address using the AWS Management Console or the command line.

To describe your Elastic IP addresses using the console

1. Open the Amazon EC2 console.
2. Click **Elastic IPs** in the navigation pane.
3. Select a filter from the Resource Attribute list to begin searching. You can use multiple filters in a single search.

To describe your Elastic IP addresses using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-addresses (AWS CLI)
- ec2-describe-addresses (Amazon EC2 CLI)
- Get-EC2Address (AWS Tools for Windows PowerShell)

Associating an Elastic IP Address with a Running Instance

You can associate an Elastic IP address to an instance using the AWS Management Console or the command line.

To associate an Elastic IP address with an instance using the console

1. Open the Amazon EC2 console.
2. Click **Elastic IPs** in the navigation pane.
3. Select an EIP and click **Associate Address**.
4. In the **Associate Address** dialog box, select the instance from the **Instance** list box and click **Associate**.

To associate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- associate-address (AWS CLI)
- ec2-associate-address (Amazon EC2 CLI)
- Register-EC2Address (AWS Tools for Windows PowerShell)

Associating an Elastic IP Address with a Different Running Instance

You can reassociate an Elastic IP address using the AWS Management Console or the command line.

To reassociate an Elastic IP address using the console

1. Open the Amazon EC2 console.
2. Click **Elastic IPs** in the navigation pane.
3. Select the EIP, and then click the **Disassociate** button.
4. Click **Yes, Disassociate** when prompted.
5. Select the EIP, and then click **Associate**.
6. In the **Associate Address** dialog box, select the new instance from the **Instance** list, and then click **Associate**.

To disassociate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

*   disassociate-address (AWS CLI)
*   ec2-disassociate-address (Amazon EC2 CLI)
*   Unregister-EC2Address (AWS Tools for Windows PowerShell)

To associate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

*   associate-address (AWS CLI)
*   ec2-associate-address (Amazon EC2 CLI)
*   Register-EC2Address (AWS Tools for Windows PowerShell)

Releasing an Elastic IP Address

If you no longer need an EIP, we recommend that you release it (the address must not be associated with an instance). You incur charges for any EIP that's allocated for use with EC2-Classic but not associated with an instance.

You can release an Elastic IP address using the AWS Management Console or the command line.

To release an Elastic IP address using the console

1. Open the Amazon EC2 console.
2. Click **Elastic IPs** in the navigation pane.
3. Select the Elastic IP address, click the **Release Address** button, and then click **Yes, Release** when prompted.

To release an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

*   release-address (AWS CLI)
*   ec2-release-address (Amazon EC2 CLI)
*   Remove-EC2Address (AWS Tools for Windows PowerShell)

Using Reverse DNS for Email Applications

If you intend to send email to third parties from an instance, we suggest you provision one or more Elastic IP addresses and provide them to us in the Request to Remove Email Sending Limitations form. AWS works with ISPs and Internet anti-spam organizations (such as Spamhaus) to reduce the chance that your email sent from these addresses will be flagged as spam.

In addition, assigning a static reverse DNS record to your Elastic IP address used to send email can help avoid having email flagged as spam by some anti-spam organizations. You can provide us with a reverse DNS record to associate with your addresses through the aforementioned form. Note that a corresponding forward DNS record (A Record) pointing to your Elastic IP address must exist before we can create your reverse DNS record.

Elastic IP Address Limit

By default, all AWS accounts are limited to 5 EIPs, because public (IPv4) Internet addresses are a scarce public resource. We strongly encourage you to use an EIP primarily for load balancing use cases, and use DNS hostnames for all other inter-node communication.

If you feel your architecture warrants additional EIPs, please complete the Amazon EC2 Elastic IP Address Request Form. We will ask you to describe your use case so that we can understand your need for additional addresses.

Elastic Network Interfaces (ENI)

An elastic network interface (ENI) is a virtual network interface that you can attach to an instance in a VPC. An ENI can include the following attributes:

- a primary private IP address

- one or more secondary private IP addresses

- one Elastic IP address per private IP address

- one public IP address, which can be auto-assigned to the network interface for eth0 when you launch an instance, but only when you create a network interface for eth0 instead of using an existing network interface

- one or more security groups

- a MAC address

- a source/destination check flag

- a description

You can create a network interface, attach it to an instance, detach it from an instance, and attach it to another instance. The attributes of a network interface follow the network interface as it is attached or detached from an instance and reattached to another instance. When you move a network interface from one instance to another, network traffic is redirected to the new instance.

Each instance in a VPC has a default network interface. The default network interface has a primary private IP address in the IP address range of its VPC. You can create and attach additional network interfaces. The maximum number of network interfaces that you can use varies by instance type. For more information, see Private IP Addresses Per ENI Per Instance Type (p. 495).

Attaching multiple network interfaces to an instance is useful when you want to:

- Create a management network.

- Use network and security appliances in your VPC.

- Create dual-homed instances with workloads/roles on distinct subnets.

- Create a low-budget, high-availability solution.

Contents

Private IP Addresses Per ENI Per Instance Type

The following table lists the maximum number of elastic network interfaces (ENI) per instance type, and the maximum number of private IP addresses per ENI. ENIs and multiple private IP addresses are only available for instances running in a VPC. For more information, see Multiple Private IP Addresses (p. 485).

Instance Type	Maximum Interfaces	IP Addresses per Interface
c1.medium	2	6
c1.xlarge	4	15
c3.large	3	10
c3.xlarge	4	15
c3.2xlarge	4	15
c3.4xlarge	8	30
c3.8xlarge	8	30
cc2.8xlarge	8	30

Instance Type	Maximum Interfaces	IP Addresses per Interface
cg1.4xlarge	8	30
cr1.8xlarge	8	30
g2.2xlarge	4	15
hi1.4xlarge	8	30
hs1.8xlarge	8	30
i2.xlarge	4	15
i2.2xlarge	4	15
i2.4xlarge	8	30
i2.8xlarge	8	30
m1.small	2	4
m1.medium	2	6
m1.large	3	10
m1.xlarge	4	15
m2.xlarge	4	15
m2.2xlarge	4	30
m2.4xlarge	8	30
m3.medium	2	6
m3.large	3	10
m3.xlarge	4	15
m3.2xlarge	4	30
r3.large	3	10
r3.xlarge	4	15
r3.2xlarge	4	15
r3.4xlarge	8	30
r3.8xlarge	8	30
t1.micro	2	2
t2.micro	2	2
t2.small	2	4
t2.medium	3	6

Creating a Management Network

You can create a management network using network interfaces. In this scenario, the secondary network interface on the instance handles public-facing traffic and the primary network interface handles back-end management traffic and is connected to a separate subnet in your VPC that has more restrictive access controls. The public facing interface, which may or may not be behind a load balancer, has an associated security group that allows access to the server from the Internet (for example, allow TCP port 80 and 443 from `0.0.0.0/0`, or from the load balancer) while the private facing interface has an associated security group allowing 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.
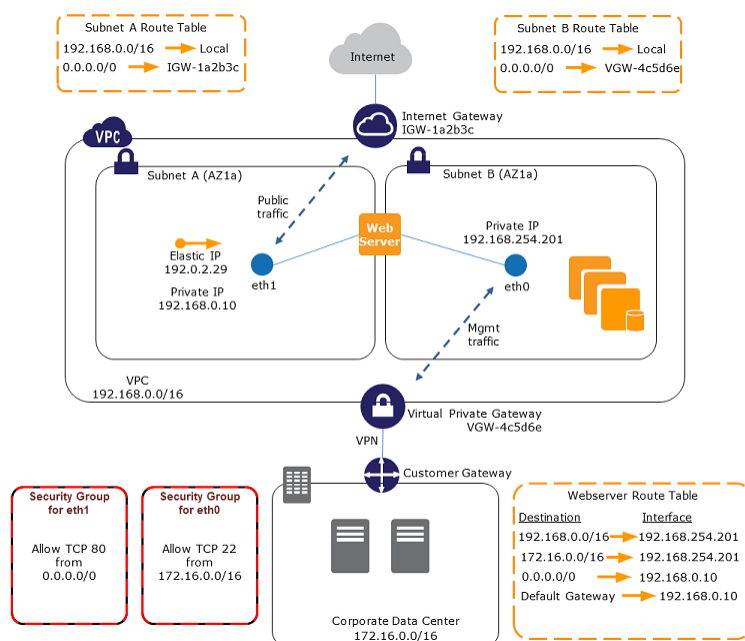
Use Network and Security Appliances in Your VPC

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 Work-loads/Roles on Distinct Subnets

You can place a network interface on each of your web servers that connects to a mid-tier network where an application server resides. The application server can also be dual-homed to a back-end network (subnet) where the database server resides. Instead of routing network packets through the dual-homed instances, each dual-homed instance receives and processes requests on the front end, initiates a connection to the back end, and then sends requests to the servers on the back-end network.

Create a Low Budget High Availability Solution

If one of your instances serving a particular function fails, its network interface can be attached to a replacement or hot standby instance pre-configured for the same role in order to rapidly recover the service. For example, you can use an ENI as your primary or secondary network interface to a critical service such as a database instance or a NAT instance. If the instance fails, you (or more likely, the code running on your behalf) can attach the ENI to a hot standby instance. Because the interface maintains its private IP addresses, Elastic IP addresses, and MAC address, network traffic will begin flowing to the standby instance as soon as you attach the ENI to the replacement instance. Users will experience a brief loss of connectivity between the time the instance fails and the time that the ENI is attached to the standby instance, but no changes to the VPC route table or your DNS server are required.

Best Practices for Configuring Network Interfaces

- You can attach a network interface to an instance when it's running (hot attach), when it's stopped (warm attach), or when the instance is being launched (cold attach).
- You can detach secondary (eth*N*) network interfaces when the instance is running or stopped. However, you can't detach the primary (eth0) interface.
- You can attach a network interface in one subnet to an instance in another subnet in the same VPC, however, both the network interface and the instance must reside in the same Availability Zone.
- When launching an instance from the CLI or API, you can specify the network interfaces to attach to the instance for both the primary (eth0) and additional network interfaces.
- Launching an instance with multiple network interfaces automatically configures interfaces, private IP addresses, and route tables on the operating system of the instance. A warm or hot attach of an additional network interface may require you to manually bring up the second interface, configure the private IP address, and modify the route table accordingly. (Instances running 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.

Configuring Your Network Interface Using ec2-net-utils

Amazon Linux AMIs may contain additional scripts installed by AWS, known as ec2-net-utils. These scripts optionally automate the configuration of your elastic network interfaces (ENIs).

Use the following command to install the package on Amazon Linux if it's not already installed, or update it if it's installed and additional updates are available:

```
$ yum install ec2-net-utils
```

The following components are part of ec2-net-utils:

udev rules (`/etc/udev/rules.d`)
   Identifies network interfaces when they are attached, detached, or reattached to a running instance, and ensures that the hotplug script runs (`53-ec2-network-interfaces.rules`). Maps the MAC address to a device name (`75-persistent-net-generator.rules`, which generates `70-persistent-net.rules`).

hotplug script
   Generates an interface configuration file suitable for use with DHCP (`/etc/sysconfig/network-scripts/ifcfg-eth`*N*). Also generates a route configuration file (`/etc/sysconfig/network-scripts/route-eth`*N*).

DHCP script
> Whenever the network interface receives a new DHCP lease, this script queries the instance metadata for Elastic IP addresses. For each Elastic IP address, it adds a rule to the routing policy database to ensure that outbound traffic from that address uses the correct network interface. It also adds each private IP address to the network interface as a secondary address.

ec2ifup eth*N*
> Extends the functionality of the standard **ifup**. After this script rewrites the configuration files `ifcfg-`eth*N* and `route-eth`*N*, it runs **ifup**.

ec2ifdown eth*N*
> Extends the functionality of the standard **ifdown**. After this script removes any rules for the network interface from the routing policy database, it runs **ifdown**.

ec2ifscan
> Checks for network interfaces that have not been configured and configures them.
>
> Note that this script isn't available in the initial release of ec2-net-utils.

To list any configuration files that were generated by ec2-net-utils, use the following command:

```
$ ls -l /etc/sysconfig/network-scripts/*-eth?
```

To disable the automation on a per-instance basis, you can add `EC2SYNC=no` to the corresponding `ifcfg-eth`*N* file. For example, use the following command to disable the automation for the eth1 interface:

```
$ sed -i -e 's/^EC2SYNC=yes/EC2SYNC=no/' /etc/sysconfig/network-scripts/ifcfg-eth1
```

If you want to disable the automation completely, you can remove the package using the following command:

```
$ yum remove ec2-net-utils
```

Creating a Network Interface

You can create a network interface using the AWS Management Console or the command line.

To create a network interface using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Click **Create Network Interface**.
4. In the **Create Network Interface** dialog box, provide the following information for the network interface, and then click **Yes, Create**.

    a. In **Description**, enter a descriptive name.
    b. In **Subnet**, select the subnet. Note that you can't move the network interface to another subnet after it's created, and you can only attach the network interface to instances in the same Availability Zone.
    c. In **Private IP**, enter the primary private IP address. If you don't specify an IP address, we'll select an available private IP address from within the selected subnet.
    d. In **Security groups**, select one or more security groups.

To create a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- create-network-interface (AWS CLI)
- ec2-create-network-interface (Amazon EC2 CLI)
- New-EC2NetworkInterface (AWS Tools for Windows PowerShell)

Deleting a Network Interface

You must first detach a network interface from an instance before you can delete it. Deleting a network interface releases all attributes associated with the network interface and releases any private IP addresses or Elastic IP addresses to be used by another instance.

You can delete a network interface using the AWS Management Console or the command line.

To delete a network interface using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Select a network interface, and then click the **Delete** button.
4. In the **Delete Network Interface** dialog box, click **Yes, Delete**.

To delete a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- delete-network-interface (AWS CLI)
- ec2-delete-network-interface (Amazon EC2 CLI)
- Remove-EC2NetworkInterface (AWS Tools for Windows PowerShell)

Viewing Details about a Network Interface

You can describe a network interface using the AWS Management Console or the command line.

To describe a network interface using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Select the network interface.
4. View the details on the **Details** tab.

To describe a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-network-interfaces (AWS CLI)
- ec2-describe-network-interfaces (Amazon EC2 CLI)
- Get-EC2NetworkInterface (AWS Tools for Windows PowerShell)

To describe a network interface attribute using the command line

You can use one of the following commands. For more information about these command line interfaces,
see Accessing Amazon EC2 (p. 3).

- describe-network-interface-attribute (AWS CLI)
- ec2-describe-network-interface-attribute (Amazon EC2 CLI)
- Get-EC2NetworkInterfaceAttribute (AWS Tools for Windows PowerShell)

Attaching a Network Interface When Launching an Instance

You can attach an additional network interface to an instance when you launch it into a VPC.

> **Note**
> If an error occurs when attaching a network interface to your instance, this causes the instance
> launch to fail.

You can attach a network interface to an instance using the AWS Management Console or the command
line.

To attach a network interface when launching an instance using the console

1. Open the Amazon EC2 console.
2. Click **Launch Instance**.
3. Choose an AMI and click its **Select** button, then choose an instance type and click **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, select a VPC from the **Network** list, and a subnet from the **Subnet** list.

   To assign a public IP address to your instance, select **Enable** from the **Auto-assign Public IP** list (if you selected a default subnet, you can leave the **Use subnet setting** option). Note that you can't assign a public IP address to your instance if you specify an existing network interface for the primary network interface (eth0) or multiple network interfaces in the next step.
5. In the **Network Interfaces** section, the console enables you specify up to 2 network interfaces (new, existing, or a combination) when you launch an instance. You can also enter a primary IP address and one or more secondary IP addresses for any new network interface. When you've finished, click **Next: Add Storage**.

   Note that you can add additional network interfaces to the instance after you launch it. The total number of network interfaces that you can attach varies by instance type. For more information, see Private IP Addresses Per ENI Per Instance Type (p. 495).
6. On the next **Add Storage** page, you can specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume), and then click **Next: Tag Instance**.
7. On the **Tag Instance** page, specify tags for the instance, such as a user-friendly name, and then click **Next: Configure Security Group**.
8. On the **Configure Security Group** page, select an existing security group or create a new one. Click **Review and Launch**.
9. On the **Review Instance Launch** page, details about the primary and additional network interface are displayed. Review the settings, and then click **Launch** to choose a key pair and launch your instance. If you're new to Amazon EC2 and haven't created any key pairs, the wizard prompts you to create one.

To attach a network interface when launching an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- run-instances (AWS CLI)
- ec2-run-instances (Amazon EC2 CLI)
- New-EC2Instance (AWS Tools for Windows PowerShell)

Attaching a Network Interface to a Stopped or Running Instance

You can attach a network interface to any of your stopped or running instances in your VPC using either the **Instances** or **Network Interfaces** page of the Amazon EC2 console, or using a command line interface.

To attach a network interface to an instance using the Instances page

1. Open the Amazon EC2 console.
2. Click **Instances** in the navigation pane.
3. Right-click the instance, and then select **Attach Network Interface**.
4. In the **Attach Network Interface** dialog box, select the network interface, and then click **Attach**.

To attach a network interface to an instance using the Network Interfaces page

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Select the network interface.
4. Click the **Attach** button.
5. In the **Attach Network Interface** dialog box, select the instance, and then click **Attach**.

To attach a network interface to an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- attach-network-interface (AWS CLI)
- ec2-attach-network-interface (Amazon EC2 CLI)
- Add-EC2NetworkInterface (AWS Tools for Windows PowerShell)

Detaching a Network Interface from an Instance

You can detach an secondary network interface at any time, using either the **Instances** or **Network Interfaces** page of the Amazon EC2 console, or using a command line interface.

To detach a network interface from an instance using the Instances page

1. Open the Amazon EC2 console.
2. Click **Instances** in the navigation pane.
3. Right-click the instance, and then select **Detach Network Interface**.
4. In the **Detach Network Interface** dialog box, select the network interface, and then click **Detach**.

To detach a network interface from an instance using the Network Interfaces page

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Select the network interface, and then click the **Detach** button.
4. In the **Detach Network Interface** dialog box, click **Yes, Detach**. If the network interface fails to detach from the instance, select **Force detachment**, and then try again.

To detach a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- detach-network-interface (AWS CLI)
- ec2-detach-network-interface (Amazon EC2 CLI)
- Dismount-EC2NetworkInterface (AWS Tools for Windows PowerShell)

Changing the Security Group of a Network Interface

You can change the security groups that are associated with a network interface. When you create the security group, be sure to specify the same VPC as the subnet for the network interface.

You can change the security group for your network interfaces using the AWS Management Console or the command line.

> **Note**
> To change security group membership for interfaces owned by other Amazon Web Services, such as Elastic Load Balancing, use the console or command line interface for that service.

To change the security group of a network interface using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Select the network interface.
4. Right-click the network interface, and then select **Change Security Groups**.
5. In the **Change Security Groups** dialog box, select the security groups to use, and then click **Save**.

To change the security group of a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- modify-network-interface-attribute (AWS CLI)
- ec2-modify-network-interface-attribute (Amazon EC2 CLI)
- Edit-EC2NetworkInterfaceAttribute (AWS Tools for Windows PowerShell)

Changing the Source/Destination Checking of a Network Interface

The Source/Destination Check attribute controls whether source/destination checking is enabled on the instance. Disabling this attribute enables an instance to handle network traffic that isn't specifically destined for the instance. For example, instances running services such as network address translation, routing, or a firewall should set this value to `disabled`. The default value is `enabled`.

You can change source/destination checking using the AWS Management Console or the command line.

To change source/destination checking for a network interface using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Right-click the network interface, and then select **Change Source/Dest Check**.
4. In the dialog box, select **Enabled** (if enabling), or **Disabled** (if disabling), and then click **Save**.

To change source/destination checking for a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

*   modify-network-interface-attribute (AWS CLI)
*   ec2-modify-network-interface-attribute (Amazon EC2 CLI)
*   Edit-EC2NetworkInterfaceAttribute (AWS Tools for Windows PowerShell)

Associating an Elastic IP Address with a Network Interface

If you have an Elastic IP address, you can associate it with one of the private IP addresses for the network interface. You can associate one Elastic IP address with each private IP address.

You can associate an Elastic IP address using the AWS Management Console or the command line.

To associate an Elastic IP address using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Right-click the network interface, and then select **Associate Address**.
4. In the **Associate Elastic IP Address** dialog box, select the Elastic IP address from the **Address** list.
5. In **Associate to private IP address**, select the private IP address to associate with the Elastic IP address.
6. Select **Allow reassociation** to allow the Elastic IP address to be associated with the specified network interface if it's currently associated with another instance or network interface, and then click **Associate Address**.

To associate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- associate-address (AWS CLI)
- ec2-associate-address (Amazon EC2 CLI)
- Register-EC2Address (AWS Tools for Windows PowerShell)

Disassociating an Elastic IP Address from a Network Interface

If the network interface has an Elastic IP address associated with it, you can disassociate the address, and then either associate it with another network interface or release it back to the address pool. Note that this is the only way to associate an Elastic IP address with an instance in a different subnet or VPC using a network interface, as network interfaces are specific to a particular subnet.

You can disassociate an Elastic IP address using the AWS Management Console or the command line.

To disassociate an Elastic IP address using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Right-click the network interface, and then select **Disassociate Address**.
4. In the **Disassociate IP Address** dialog box, click **Yes, Disassociate**.

To disassociate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- disassociate-address (AWS CLI)
- ec2-disassociate-address (Amazon EC2 CLI)
- Unregister-EC2Address (AWS Tools for Windows PowerShell)

Changing Termination Behavior for a Network Interface

You can set the termination behavior for a network interface attached to an instance so that it is automatically deleted when you delete the instance it's attached to.

> **Note**
> By default, network interfaces that are automatically created and attached to instances using the AWS Management Console are set to terminate when the instance terminates. However, network interfaces created using the command line interface aren't set to terminate when the instance terminates.

You can change the terminating behavior for a network interface using the AWS Management Console or the command line.

To change the termination behavior for a network interface using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Right-click the network interface, and then select **Change Termination Behavior**.

4. In the **Change Termination Behavior** dialog box, select the **Delete on termination** check box if you want the network interface to be deleted when you terminate an instance.

To change the termination behavior for a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- modify-network-interface-attribute (AWS CLI)
- ec2-modify-network-interface-attribute (Amazon EC2 CLI)
- Edit-EC2NetworkInterfaceAttribute (AWS Tools for Windows PowerShell)

Adding or Editing a Description for a Network Interface

You can change the description for a network interface using the AWS Management Console or the command line.

To change the description for a network interface using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Right-click the network interface, and then select **Change Description**.
4. In the **Change Description** dialog box, enter a description for the network interface, and then click **Save**.

To change the description for a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- modify-network-interface-attribute (AWS CLI)
- ec2-modify-network-interface-attribute (Amazon EC2 CLI)
- Edit-EC2NetworkInterfaceAttribute (AWS Tools for Windows PowerShell)

Adding or Editing Tags for a Network Interface

Tags are metadata that you can add to a network interface. Tags are private and are only visible to your account. Each tag consists of a key and an optional value. For more information about tags, see Tagging Your Amazon EC2 Resources (p. 610).

You can tag a resource using the AWS Management Console or the command line.

To add or edit tags for a network interface using the console

1. Open the Amazon EC2 console.
2. Click **Network Interfaces** in the navigation pane.
3. Select the network interface.
4. In the details pane, click the **Tags** tab, and then click **Add/Edit Tags**.
5. In the **Add/Edit Tags** dialog, click **Create Tag** for each tag you want to create, and enter a key and optional value. When you're done, click **Save**.

To add or edit tags for a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- create-tags (AWS CLI)
- ec2-create-tags (Amazon EC2 CLI)
- New-EC2Tag (AWS Tools for Windows PowerShell)

Enabling Enhanced Networking on Linux Instances in a VPC

With C3, R3, and I2 instances, you can enable enhanced networking capabilities. Amazon EC2 supports enhanced networking capabilities using single root I/O virtualization (SR-IOV). Enabling enhanced networking on your instance results in higher performance (packets per second), lower latency, and lower jitter.

To enable enhanced networking on your instance, you must ensure that its kernel has the `ixgbevf` module installed and that you set the `sriovNetSupport` attribute for the instance. For the best performance, we recommend that the `ixgbevf` module is version 2.14.2 or higher.

> **Important**
> The latest Amazon Linux HVM AMIs have the module required for enhanced networking installed and have the required attribute set. Therefore, if you launch an Amazon EBS–backed C3, R3, or I2 instance using a current Amazon Linux HVM AMI, enhanced networking is already enabled for your instance.

Contents

- Requirements (p. 507)
- Testing Whether Enhanced Networking Is Enabled (p. 508)
- Enabling Enhanced Networking on Amazon Linux (p. 510)
- Enabling Enhanced Networking on Ubuntu (p. 512)
- Enabling Enhanced Networking on Other Linux Distributions (p. 515)
- Troubleshooting Connectivity Issues (p. 517)

Note that you can get directions for Windows from Enabling Enhanced Networking on Windows Instances in a VPC in the *Amazon EC2 User Guide for Microsoft Windows Instances*.

Requirements

Before enabling enhanced networking, make sure you do the following:

- Launch the instance from an HVM AMI with a minimum Linux kernel version of 2.6.32.
- Launch the instance using one of the following instance types: `c3.large`, `c3.xlarge`, `c3.2xlarge`, `c3.4xlarge`, `c3.8xlarge`, `i2.xlarge`, `i2.2xlarge`, `i2.4xlarge`, `i2.8xlarge`, `r3.large`, `r3.xlarge`, `r3.2xlarge`, `r3.4xlarge`, or `r3.8xlarge`. For more information about instance types, see Amazon EC2 Instances.
- Launch the instance in a VPC. (You can't enabled enhanced networking if the instance is in EC2-Classic.)
- Install and configure either the AWS CLI or Amazon EC2 CLI tools to any computer you choose, preferably your local desktop or laptop. For more information, see Accessing Amazon EC2 (p. 3). If

you choose the Amazon EC2 CLI tools, install version 1.6.12.0 or later. You can use the ec2-version command to verify the version of your CLI tools.

- If you have important data on the instance that you want to preserve, you should back that data up now by creating a snapshot for Amazon EBS-backed instances or creating an AMI for instance store-backed instances. Updating kernels and kernel modules as well as enabling the `sriovNetSupport` attribute may make incompatible instances or operating systems unreachable; if you have a recent backup, your data will still be retained if this happens.

Testing Whether Enhanced Networking Is Enabled

To test whether enhanced networking is already enabled, verify that the `ixgbevf` module is installed on your instance and that the `sriovNetSupport` attribute is set. If your instance satisfies these two conditions, then the **ethtool -i eth*n*** command should show that the module is in use on the network interface.

Kernel Module (ixgbevf)

To verify that the `ixgbevf` module is installed and that the version is compatible with enhanced networking, use the **modinfo** command as follows:

```
[ec2-user ~]$ modinfo ixgbevf
filename:       /lib/modules/3.10.48-55.140.amzn1.x86_64/ker
nel/drivers/amazon/ixgbevf/ixgbevf.ko
version:        2.14.2
license:        GPL
description:    Intel(R) 82599 Virtual Function Driver
author:         Intel Corporation, <linux.nics@intel.com>
srcversion:     50CBF6F36B99FE70E56C95A
alias:          pci:v00008086d00001515sv*sd*bc*sc*i*
alias:          pci:v00008086d000010EDsv*sd*bc*sc*i*
depends:
intree:         Y
vermagic:       3.10.48-55.140.amzn1.x86_64 SMP mod_unload modversions
parm:           InterruptThrottleRate:Maximum interrupts per second, per vector,
 (956-488281, 0=off, 1=dynamic), default 1 (array of int)
```

In the above Amazon Linux case, the `ixgbevf` module is already installed and it is at the minimum recommended version (2.14.2).

```
ubuntu:~$ modinfo ixgbevf
filename:       /lib/modules/3.13.0-29-generic/kernel/drivers/net/ethernet/in
tel/ixgbevf/ixgbevf.ko
version:        2.11.3-k
license:        GPL
description:    Intel(R) 82599 Virtual Function Driver
author:         Intel Corporation, <linux.nics@intel.com>
srcversion:     0816EA811025C8062A9C269
alias:          pci:v00008086d00001515sv*sd*bc*sc*i*
alias:          pci:v00008086d000010EDsv*sd*bc*sc*i*
depends:
intree:         Y
vermagic:       3.13.0-29-generic SMP mod_unload modversions
signer:         Magrathea: Glacier signing key
sig_key:        66:02:CB:36:F1:31:3B:EA:01:C4:BD:A9:65:67:CF:A7:23:C9:70:D8
sig_hashalgo:   sha512
parm:           debug:Debug level (0=none,...,16=all) (int)
```

In the above Ubuntu instance, the module is installed, but the version is 2.11.3-k, which does not have all of the latest bug fixes that the recommended version 2.14.2 does. In this case, the `ixgevf` module would work, but a newer version can still be installed and loaded on the instance for the best experience.

Instance Attribute (sriovNetSupport)

To check whether an instance has the enhanced networking attribute set, use one of the following commands:

- describe-instance-attribute (AWS CLI)

```
$ aws ec2 describe-instance-attribute --instance-id instance_id --attribute
sriovNetSupport
```

If the enhanced networking attribute isn't set, `SriovNetSupport` is empty. Otherwise, it is set as follows:

```
"SriovNetSupport": {
    "Value": "simple"
},
```

- ec2-describe-instance-attribute (Amazon EC2 CLI)

```
$ ec2-describe-instance-attribute instance_id --sriov
```

If the enhanced networking attribute isn't set, you'll see no output for this command. Otherwise, the output is as follows:

```
sriovNetSupport instance_id    simple
```

Image Attribute (sriovNetSupport)

To check whether an AMI already has the enhanced networking attribute set, use one of the following commands:

- describe-image-attribute (AWS CLI)

```
$ aws ec2 describe-image-attribute --image-id ami_id --attribute sriovNetSup
port
```

> **Note**
> This command only works for images that you own. You receive an `AuthFailure` error for images that do not belong to your account.

If the enhanced networking attribute isn't set, `SriovNetSupport` is empty. Otherwise, it is set as follows:

```
"SriovNetSupport": {
    "Value": "simple"
},
```

- ec2-describe-image-attribute (Amazon EC2 CLI)

```
$ ec2-describe-image-attribute ami_id --sriov
```

> **Note**
> This command only works for images that you own. You will receive an `AuthFailure` error
> for images that do not belong to your account.

If the enhanced networking attribute isn't set, you'll see no output for this command. Otherwise, the
output is as follows:

```
sriovNetSupport ami_id          simple
```

Network Interface Driver

Use the following command to verify that the module is being used on a particular interface, substituting
the interface name that you wish to check. If you are using a single interface (default), it will be `eth0`.

```
[ec2-user ~]$ ethtool -i eth0
driver: vif
version:
firmware-version:
bus-info: vif-0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

In the above case, the `ixgbevf` module is not loaded, because the listed driver is `vif`.

```
[ec2-user ~]$ ethtool -i eth0
driver: ixgbevf
version: 2.14.2
firmware-version: N/A
bus-info: 0000:00:03.0
supports-statistics: yes
supports-test: yes
supports-eeprom-access: no
supports-register-dump: yes
supports-priv-flags: no
```

In this case, the `ixgbevf` module is loaded and at the minimum recommended version. This instance
has enhanced networking properly configured.

Enabling Enhanced Networking on Amazon Linux

The latest Amazon Linux HVM AMIs have the module required for enhanced networking installed and
have the required attribute set. Therefore, if you launch an Amazon EBS–backed C3, R3, or I2 instance
using a current Amazon Linux HVM AMI, enhanced networking is already enabled for your instance. For
more information, see Testing Whether Enhanced Networking Is Enabled (p. 508).

If you launched your instance using an older Amazon Linux AMI and it does not have enhanced networking
enabled already, use the following procedure to enable enhanced networking.

To enable enhanced networking (EBS-backed instances)

1. Connect to your instance.

2. From the instance, run the following command to update your instance with the newest kernel and kernel modules, including `ixgbevf`:

    ```
    [ec2-user ~]$ sudo yum update
    ```

3. From your local computer, reboot your instance using the Amazon EC2 console or one of the following commands: reboot-instances (AWS CLI) or ec2-reboot-instances (Amazon EC2 CLI).

4. Connect to your instance again and verify that the `ixgbevf` module is installed and at the minimum recommended version using the **modinfo ixgbevf** command from Testing Whether Enhanced Networking Is Enabled (p. 508).

5. From your local computer, stop the instance using the Amazon EC2 console or one of the following commands: stop-instances (AWS CLI) or ec2-stop-instances (Amazon EC2 CLI).

    > **Important**
    > If you are using an instance store-backed instance, you can't stop the instance. Instead, proceed to To enable enhanced networking (instance store-backed instances) (p. 511).

6. From your local computer, enable the enhanced networking attribute using one of the following commands.

    > **Warning**
    > There is no way to disable the enhanced networking attribute after you've enabled it.

    > **Warning**
    > Enhanced networking is supported only for HVM instances. Enabling enhanced networking with a PV instance can make it unreachable. Setting this attribute without the proper module or module version can also make your instance unreachable.

    *   modify-instance-attribute (AWS CLI)

        ```
        $ aws ec2 modify-instance-attribute --instance-id instance_id --sriov-net-
        support simple
        ```

    *   ec2-modify-instance-attribute (Amazon EC2 CLI)

        ```
        $ ec2-modify-instance-attribute instance_id --sriov simple
        ```

7. (Optional) Create an AMI from the instance, as described in Creating an Amazon EBS-Backed Linux AMI (p. 74) . The AMI inherits the enhanced networking attribute from the instance. Therefore, you can use this AMI to launch another C3, R3, or I2 instance with the enhanced networking enabled by default.

8. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: start-instances (AWS CLI) or ec2-start-instances (Amazon EC2 CLI).

9. Connect to your instance and verify that the `ixgbevf` module is installed and loaded on your network interface using the **ethtool -i eth*n*** command from Testing Whether Enhanced Networking Is Enabled (p. 508).

To enable enhanced networking (instance store-backed instances)

If your instance is an instance store–backed instance, follow Step 1 (p. 511) through Step 4 (p. 511) in the previous procedure, and then create a new AMI as described in Creating an Instance Store-Backed Linux AMI (p. 77). Be sure to enable the enhanced networking attribute when you register the AMI.

- register-image (AWS CLI)

```
$ aws ec2 register-image --sriov-net-support simple ...
```

- ec2-register (Amazon EC2 CLI)

```
$ ec2-register --sriov simple ...
```

Enabling Enhanced Networking on Ubuntu

The following procedure provides the general steps that you'll take when enabling enhanced networking on an Ubuntu instance.

To enable enhanced networking on Ubuntu (EBS-backed instances)

1. Connect to your instance.
2. Update the package cache and packages.

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y
```

> **Important**
> If during the update process you are prompted to install grub, use /dev/xvda to install grub onto, and then choose to keep the current version of /boot/grub/menu.lst.

3. Install the **dkms** package so that your ixgbevf module is rebuilt every time your kernel is updated.

```
ubuntu:~$ sudo apt-get install -y dkms
```

4. Download the latest source for the ixgbevf module on your instance from Sourceforge at http://sourceforge.net/projects/e1000/files/ixgbevf%20stable/. At the time of this writing, that version was 2.14.2. If the latest version available is not 2.14.2, substitute your version of ixgbevf in the following commands.

```
ubuntu:~$ wget "sourceforge.net/projects/e1000/files/ixgbevf
stable/2.14.2/ixgbevf-2.14.2.tar.gz"
```

5. Decompress and unarchive the ixgbevf package.

```
ubuntu:~$ tar -xzf ixgbevf-2.14.2.tar.gz
```

6. Move the ixgbevf package to the /usr/src/ directory so **dkms** can find it and build it for each kernel update.

```
ubuntu:~$ sudo mv ixgbevf-2.14.2 /usr/src/
```

7. Create the **dkms** configuration file with the following values, substituting your version of ixgbevf.

   a. Create the file.

   ```
   ubuntu:~$ sudo touch /usr/src/ixgbevf-2.14.2/dkms.conf
   ```

   b. Edit the file and add the following values.

   ```
   ubuntu:~$ sudo vim /usr/src/ixgbevf-2.14.2/dkms.conf
   PACKAGE_NAME="ixgbevf"
   PACKAGE_VERSION="2.14.2"
   CLEAN="cd src/; make clean"
   MAKE="cd src/; make BUILD_KERNEL=${kernelver}"
   BUILT_MODULE_LOCATION[0]="src/"
   BUILT_MODULE_NAME[0]="ixgbevf"
   DEST_MODULE_LOCATION[0]="/updates"
   DEST_MODULE_NAME[0]="ixgbevf"
   AUTOINSTALL="yes"
   ```

8. Add, build, and install the ixgbevf module on your instance with **dkms**.

   a. Add the module to **dkms**.

   ```
   ubuntu:~$ sudo dkms add -m ixgbevf -v 2.14.2
   ```

   b. Build the module with **dkms**.

   ```
   ubuntu:~$ sudo dkms build -m ixgbevf -v 2.14.2
   ```

   c. Install the module with **dkms**.

   ```
   ubuntu:~$ sudo dkms install -m ixgbevf -v 2.14.2
   ```

9. Rebuild the initramfs so the correct module is loaded at boot time.

   ```
   ubuntu:~$ sudo update-initramfs -c -k all
   ```

10. Verify that the ixgbevf module is installed and at the minimum recommended version using the
    modinfo ixgbevf command from Testing Whether Enhanced Networking Is Enabled (p. 508).

    ```
    ubuntu:~$ modinfo ixgbevf
    filename:       /lib/modules/3.13.0-29-generic/updates/dkms/ixgbevf.ko
    version:        2.14.2
    license:        GPL
    description:    Intel(R) 82599 Virtual Function Driver
    author:         Intel Corporation, <linux.nics@intel.com>
    srcversion:     50CBF6F36B99FE70E56C95A
    alias:          pci:v00008086d00001515sv*sd*bc*sc*i*
    ```

```
alias:          pci:v00008086d000010EDsv*sd*bc*sc*i*
depends:
vermagic:       3.13.0-29-generic SMP mod_unload modversions
parm:           InterruptThrottleRate:Maximum interrupts per second, per
vector, (956-488281, 0=off, 1=dynamic), default 1 (array of int)
```

11. From your local computer, stop the instance using the Amazon EC2 console or one of the following commands: stop-instances (AWS CLI) or ec2-stop-instances (Amazon EC2 CLI).

    Important
    If you are using an instance store-backed instance, you can't stop the instance. Instead, proceed to To enable enhanced networking on Ubuntu (instance store-backed in-stances) (p. 514).

12. From your local computer, enable the enhanced networking attribute using one of the following commands. Note that there is no way to disable the networking attribute after you've enabled it.

    Warning
    Enhanced networking is supported only for HVM instances. Enabling enhanced networking with a PV instance can make it unreachable. Setting this attribute without the proper module or module version can also make your instance unreachable.

    - modify-instance-attribute (AWS CLI)

      ```
      $ aws ec2 modify-instance-attribute --instance-id instance_id --sriov-net-
      support simple
      ```

    - ec2-modify-instance-attribute (Amazon EC2 CLI)

      ```
      $ ec2-modify-instance-attribute instance_id --sriov simple
      ```

13. (Optional) Create an AMI from the instance, as described in Creating an Amazon EBS-Backed Linux AMI (p. 74) . The AMI inherits the enhanced networking attribute from the instance. Therefore, you can use this AMI to launch a C3, R3, or I2 instance with the enhanced networking enabled by default.

14. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: start-instances (AWS CLI) or ec2-start-instances (Amazon EC2 CLI).

15. (Optional) Connect to your instance and verify that the module is installed.

To enable enhanced networking on Ubuntu (instance store-backed instances)

If your instance is an instance store-backed instance, follow Step 1 (p. 512) through Step 10 (p. 513) in the previous procedure, and then create a new AMI as described in Creating an Instance Store-Backed Linux AMI (p. 77). Be sure to enable the enhanced networking attribute when you register the AMI.

    Warning
    Enhanced networking is supported only for HVM instances. Enabling enhanced networking with a PV instance can make it unreachable. Setting this attribute without the proper module or module version can also make your instance unreachable.

- register-image (AWS CLI)

  ```
  $ aws ec2 register-image --sriov-net-support simple ...
  ```

- ec2-register (Amazon EC2 CLI)

```
$ ec2-register --sriov simple ...
```

Enabling Enhanced Networking on Other Linux Distributions

The following procedure provides the general steps that you'll take when enabling enhanced networking on a Linux distribution other than Amazon Linux or Ubuntu. For more information, such as detailed syntax for commands, file locations, or package and tool support, see the specific documentation for your Linux distribution.

To enable enhanced networking on Linux (EBS-backed instances)

1. Connect to your instance.
2. Download the latest source for the `ixgbevf` module on your instance from Sourceforge at http://sourceforge.net/projects/e1000/files/ixgbevf%20stable/. At the time of this writing, that version was 2.14.2; this is the minimum version recommended for enhanced networking.
3. Compile and install the `ixgbevf` module on your instance.

   If your distribution supports **dkms**, then you should consider configuring **dkms** to recompile the `ixgbevf` module whenever your system's kernel is updated. If your distribution does not support **dkms** natively, you can find it in the EPEL repository (https://fedoraproject.org/wiki/EPEL) for Red Hat Enterprise Linux variants, or you can download the software at http://linux.dell.com/dkms/. Use Step 6 (p. 512) through Step 8 (p. 513) in To enable enhanced networking on Ubuntu (EBS-backed instances) (p. 512) for help configuring **dkms**.

   > **Warning**
   > If you compile the `ixgbevf` module for your current kernel and then upgrade your kernel without rebuilding the driver for the new kernel, your system may revert to the distribution-specific `ixgbevf` module at the next reboot, which could make your system unreachable if the distribution-specific version is incompatible with enhanced networking.

4. Run the **sudo depmod** command to update module dependencies.
5. Update the `initramfs` on your instance to ensure that the new module loads at boot time.
6. Determine if your system uses predictable network interface names by default. Systems that use **systemd** or **udev** versions 197 or greater can rename Ethernet devices and they do not guarantee that a single network interface will be named `eth0`. This behavior can cause problems connecting to your instance. For more information on predictable network interface names, and to see other configuration options, go to http://www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames/.

   a. You can check the **systemd** or **udev** versions on RPM-based systems with the following command:

   ```
   [ec2-user ~]$ rpm -qa | grep -e '^systemd-[0-9]\+\|^udev-[0-9]\+'
   systemd-208-11.el7_0.2.x86_64
   ```

   In the above Red Hat 7 example, the **systemd** version is 208, so predictable network interface names must be disabled.

   b. Disable predictable network interface names by adding the `net.ifnames=0` option to the `GRUB_CMDLINE_LINUX` line in `/etc/default/grub`.

```
[ec2-user ~]$ sudo sed -i '/^GRUB\_CMDLINE\_LINUX/s/\"$/\ net\.if
names\=0\"/' /etc/default/grub
```

c.   Rebuild the grub configuration file.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7.   From your local computer, stop the instance using the Amazon EC2 console or one of the following
     commands: stop-instances (AWS CLI) or ec2-stop-instances (Amazon EC2 CLI).

     Important
     If you are using an instance store-backed instance, you can't stop the instance. Instead,
     proceed to To enabled enhanced networking (instance store–backed instances) (p. 516)

8.   From your local computer, enable the enhanced networking attribute using one of the following
     commands. Note that there is no way to disable the networking attribute after you've enabled it.

     Warning
     Enhanced networking is supported only for HVM instances. Enabling enhanced networking
     with a PV instance can make it unreachable. Setting this attribute without the proper module
     or module version can also make your instance unreachable.

     • modify-instance-attribute (AWS CLI)

```
$ aws ec2 modify-instance-attribute --instance-id instance_id --sriov-net-
support simple
```

     • ec2-modify-instance-attribute (Amazon EC2 CLI)

```
$ ec2-modify-instance-attribute instance_id --sriov simple
```

9.   (Optional) Create an AMI from the instance, as described in Creating an Amazon EBS-Backed Linux
     AMI (p. 74) . The AMI inherits the enhanced networking attribute from the instance. Therefore, you
     can use this AMI to launch a C3, R3, or I2 instance with the enhanced networking enabled by default.
10. From your local computer, start the instance using the Amazon EC2 console or one of the following
     commands: start-instances (AWS CLI) or ec2-start-instances (Amazon EC2 CLI).
11. (Optional) Connect to your instance and verify that the module is installed.

To enabled enhanced networking (instance store–backed instances)

If your instance is an instance store–backed instance, follow Step 1 (p. 515) through Step 5 (p. 515) in the
previous procedure, and then create a new AMI as described in Creating an Instance Store-Backed Linux
AMI (p. 77). Be sure to enable the enhanced networking attribute when you register the AMI.

     Warning
     Enhanced networking is supported only for HVM instances. Enabling enhanced networking with
     a PV instance can make it unreachable. Setting this attribute without the proper module or
     module version can also make your instance unreachable.

• register-image (AWS CLI)

```
$ aws ec2 register-image --sriov-net-support simple ...
```

- ec2-register (Amazon EC2 CLI)

```
$ ec2-register --sriov simple ...
```

Troubleshooting Connectivity Issues

If you lose connectivity while enabling enhanced networking, the `ixgbevf` module might be incompatible with the kernel. Try installing the version of the `ixgbevf` module included with the distribution of Linux for your instance.

If you enable enhanced networking for a PV instance or AMI, this can make your instance unreachable.

Storage

Amazon EC2 provides you with flexible, cost effective, and easy-to-use data storage options for your in-stances. 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:

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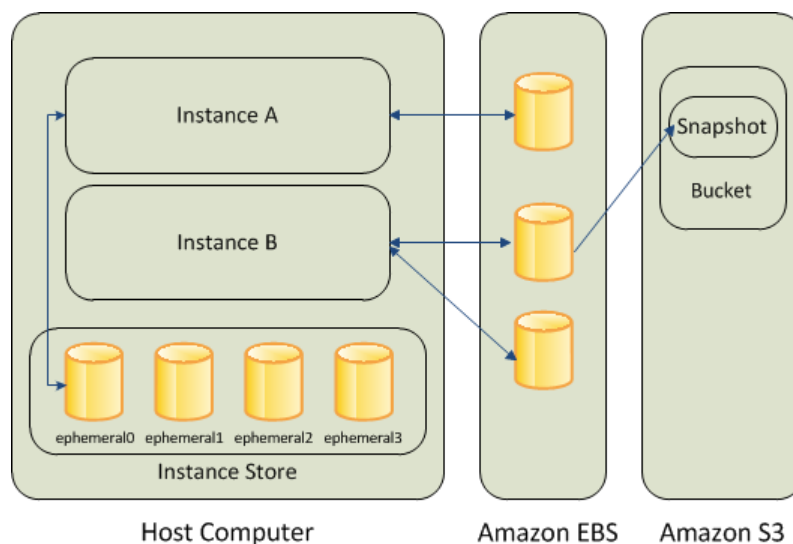
Amazon EBS

Amazon EBS provides durable, block-level storage volumes that you can attach to a running Amazon EC2 instance. You can use Amazon EBS as a primary storage device for data that requires frequent and

granular updates. For example, Amazon EBS is the recommended storage option when you run a database on an instance.

An Amazon EBS volume behaves like a raw, unformatted, external block device that you can attach to a single instance. The volume persists independently from the running life of an Amazon EC2 instance. After an EBS volume is attached to an instance, you can use it like any other physical hard drive. As illustrated in the previous figure, multiple volumes can be attached to an instance. You can also detach an EBS volume from one instance and attach it to another instance. Amazon EBS volumes can also be created as encrypted volumes using the Amazon EBS encryption feature. For more information, see Amazon EBS Encryption (p. 564).

To keep a backup copy of your data, you can create a *snapshot* of an EBS volume, which is stored in Amazon S3. You can create a new Amazon EBS volume from a snapshot, and attach it to another instance. For more information, see Amazon Elastic Block Store (Amazon EBS) (p. 519).

Amazon EC2 Instance Store

Many Amazon EC2 instances can access storage from disks that are physically attached to the host computer. This disk storage is referred to as *instance store*. Instance store provides temporary block-level storage for Amazon EC2 instances. The data on an instance store volume persists only during the life of the associated Amazon EC2 instance; if you stop or terminate an instance, any data on instance store volumes is lost. For more information, see Amazon EC2 Instance Store (p. 581).

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. 589).

Adding Storage

Every time you launch an instance from an AMI, a root storage device is created for that instance. The root storage device contains all the information necessary to boot the instance. You can specify storage volumes in addition to the root device volume when you create an AMI or launch an instance using *block device mapping*. For more information, see Block Device Mapping (p. 591).

You can also attach EBS volumes to a running instance. For more information, see Attaching an Amazon EBS Volume to an Instance (p. 529).

Amazon Elastic Block Store (Amazon EBS)

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with Amazon EC2 instances. Amazon EBS volumes are highly available and reliable storage volumes that can be attached to any running instance that is in the same Availability Zone. Amazon EBS volumes that are attached to an Amazon EC2 instance are exposed as storage volumes that persist independently from the life of the instance. With Amazon EBS, you pay only for what you use. For more information about Amazon EBS pricing, see the Projecting Costs section of the Amazon Elastic Block Store page.

Amazon EBS is recommended when data changes frequently and requires long-term persistence. Amazon EBS volumes are particularly well-suited for use as the primary storage for file systems, databases, or for any applications that require fine granular updates and access to raw, unformatted, block-level storage. Amazon EBS is particularly helpful for database-style applications that frequently encounter many random reads and writes across the data set.

For simplified data encryption, you can launch your Amazon EBS volumes as encrypted volumes. Amazon EBS encryption offers you a simple encryption solution for your EBS volumes without the need for you to build, manage, and secure your own key management infrastructure. When you create an encrypted EBS volume and attach it to a supported instance type, data stored at rest on the volume, disk I/O, and snapshots created from the volume are all encrypted. The encryption occurs on the servers that hosts EC2 instances, providing encryption of data-in-transit from EC2 instances to EBS storage. For more information, see Amazon EBS Encryption (p. 564).

Amazon EBS encryption uses AWS Key Management Service (AWS KMS) master keys when creating encrypted volumes and any snapshots created from your encrypted volumes. The first time you create an encrypted Amazon EBS volume in a region, a default master key is created for you automatically. This key is used for Amazon EBS encryption unless you select a Customer Master Key (CMK) that you created separately using the AWS Key Management Service. Creating your own CMK gives you more flexibility, including the ability to create, rotate, disable, define access controls, and audit the encryption keys used to protect your data. For more information, see the AWS Key Management Service Developer Guide.

You can attach multiple volumes to the same instance within the limits specified by your AWS account. Your account has a limit on the number of Amazon EBS volumes that you can use, and the total storage available to you. For more information about these limits, and how to request an increase in your limits, see Request to Increase the Amazon EBS Volume Limit.

Contents

Features of Amazon EBS

- You can create Amazon EBS volumes from 1 GiB to 1 TiB in size. You can mount these volumes as devices on your Amazon EC2 instances. You can mount multiple volumes on the same instance, but each volume can be attached to only one instance at a time. For more information, see Creating an Amazon EBS Volume (p. 526).
- With General Purpose (SSD) volumes, your volume receives a base performance of 3 IOPS/GiB, with the ability to burst to 3,000 IOPS for extended periods of time. General Purpose (SSD) volumes are ideal for a broad range of use cases such as boot volumes, small and medium size databases, and development and test environments. For more information, see General Purpose (SSD) Volumes (p. 524).
- With Provisioned IOPS (SSD) volumes, you can provision a specific level of I/O performance, up to 4000 IOPS per volume. This allows you to predictably scale to thousands of IOPS per EC2 instance. For more information, see Provisioned IOPS (SSD) Volumes (p. 525).
- Amazon EBS volumes behave like raw, unformatted block devices. You can create a file system on top of these volumes, or use them in any other way you would use a block device (like a hard drive). For more information on creating file systems and mounting volumes, see Making an Amazon EBS Volume Available for Use (p. 531).
- You can use encrypted Amazon EBS volumes to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications. For more information, see Amazon EBS Encryption (p. 564).
- You can create point-in-time snapshots of Amazon EBS volumes, which are persisted to Amazon S3. Snapshots protect data for long-term durability, and they can be used as the starting point for new Amazon EBS volumes. The same snapshot can be used to instantiate as many volumes as you wish. These snapshots can be copied across AWS regions. For more information, see Amazon EBS Snapshots (p. 559).

- Amazon EBS volumes are created in a specific Availability Zone, and can then be attached to any in-stances in that same Availability Zone. To make a volume available outside of the Availability Zone, you can create a snapshot and restore that snapshot to a new volume anywhere in that region. You can copy snapshots to other regions and then restore them to new volumes there, making it easier to leverage multiple AWS regions for geographical expansion, data center migration, and disaster recovery. For more information, see Creating an Amazon EBS Snapshot (p. 560), Restoring an Amazon EBS Volume from a Snapshot (p. 527), and Copying an Amazon EBS Snapshot (p. 561).

- A large repository of public data set snapshots can be restored to Amazon EBS volumes and seamlessly integrated into AWS cloud-based applications. For more information, see Using Public Data Sets (p. 602).

- Performance metrics, such as bandwidth, throughput, latency, and average queue length, are available through the AWS Management Console. These metrics, provided by Amazon CloudWatch, allow you to monitor the performance of your volumes to make sure that you are providing enough performance for your applications without paying for resources you don't need. For more information, see Amazon EBS Volume Performance (p. 567).

Amazon EBS Volumes

An Amazon EBS volume is a durable, block-level storage device that you can attach to a single EC2 in-stance. You can use Amazon EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application. Amazon EBS volumes persist independently from the running life of an EC2 instance. After a volume is attached to an instance, you can use it like any other physical hard drive. Amazon EBS provides the following volume types: General Purpose (SSD), Provisioned IOPS (SSD), and Magnetic. They differ in performance character-istics and price, allowing you to tailor your storage performance and cost to the needs of your applications. For more information, see Amazon EBS Volume Types (p. 523).

Contents

Benefits of Using Amazon EBS Volumes

Data Availability

When you create an Amazon EBS volume in an Availability Zone, it is automatically replicated within that zone to prevent data loss due to failure of any single hardware component. After you create a volume, you can attach it to any Amazon EC2 instance in the same Availability Zone. After you attach a volume, it appears as a native block device similar to a hard drive or other physical device. At that point, the instance can interact with the volume just as it would with a local drive; the instance can format the Amazon EBS volume with a file system, such as ext3, and then install applications.

An Amazon EBS volume can be attached to only one instance at a time within the same Availability Zone. However, multiple volumes can be attached to a single instance. If you attach multiple volumes to a device that you have named, you can stripe data across the volumes for increased I/O and throughput performance.

You can get monitoring data for your Amazon EBS volumes at no additional charge (this includes data for the root device volumes for Amazon EBS-backed instances). For more information, see Monitoring Volumes with CloudWatch (p. 534).

Data Persistence

An Amazon EBS volume is off-instance storage that can persist independently from the life of an instance. You continue to pay for the volume usage as long as the data persists.

By default, Amazon EBS volumes that are attached to a running instance automatically detach from the instance with their data intact when that instance is terminated. The volume can then be reattached to a new instance, enabling quick recovery. If you are using an Amazon EBS-backed instance, you can stop and restart that instance without affecting the data stored in the attached volume. The volume remains attached throughout the stop-start cycle. This enables you to process and store the data on your volume indefinitely, only using the processing and storage resources when required. The data persists on the volume until the volume is deleted explicitly. After a volume is deleted, it can't be attached to any instance. The physical block storage used by deleted Amazon EBS volumes is overwritten with zeroes before it is allocated to another account. If you are dealing with sensitive data, you should consider encrypting your data manually or storing the data on a volume that is enabled with Amazon EBS encryption. For more information, see Amazon EBS Encryption (p. 564).

By default, Amazon EBS volumes that are created and attached to an instance at launch are deleted when that instance is terminated. You can modify this behavior by changing the value of the flag `Delete-OnTermination` to `false` when you launch the instance. This modified value causes the volume to persist even after the instance is terminated, and enables you to attach the volume to another instance.

Data Encryption

For simplified data encryption, you can create encrypted Amazon EBS volumes with the Amazon EBS encryption feature. You can use encrypted Amazon EBS volumes to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications. Amazon EBS encryption uses 256-bit Advanced Encryption Standard algorithms (AES-256) and an Amazon-managed key infrastructure. The encryption occurs on the server that hosts the Amazon EC2 instance, providing encryption of data-in-transit from the EC2 instance to EBS storage. For more information, see Amazon EBS Encryption (p. 564).

Amazon EBS encryption uses AWS Key Management Service (AWS KMS) master keys when creating encrypted volumes and any snapshots created from your encrypted volumes. The first time you create an encrypted Amazon EBS volume in a region, a default master key is created for you automatically. This key is used for Amazon EBS encryption unless you select a Customer Master Key (CMK) that you created separately using the AWS Key Management Service. Creating your own CMK gives you more flexibility, including the ability to create, rotate, disable, define access controls, and audit the encryption keys used to protect your data. For more information, see the AWS Key Management Service Developer Guide.

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 be attached to a running instance in order to take a snapshot. As you continue to write data to a volume, you can periodically create a snapshot of the volume to use as a baseline for new volumes. These snapshots can be used to create multiple new Amazon EBS volumes, expand the size of a volume, or move volumes across Availability Zones. Snapshots of encrypted Amazon EBS volumes are automatically encrypted.

When you create a new volume from a snapshot, it's an exact copy of the original volume at the time the snapshot was taken. Amazon EBS volumes that are restored from encrypted snapshots are automatically encrypted. By optionally specifying a different volume size or a different Availability Zone, you can use this functionality to increase the size of an existing volume or to create duplicate volumes in new Availability Zones. The snapshots can be shared with specific AWS accounts or made public. When you create snapshots, you incur charges in Amazon S3 based on the volume's total size. For a successive snapshot of the volume, you are only charged for any additional data beyond the volume's original size.

Amazon EBS snapshots are incremental backups, meaning that only the blocks on the volume that have changed after your most recent snapshot are saved. If you have a volume with 100 GiB of data, but only 5 GiB of data have changed since your last snapshot, only the 5 GiB of modified data is written to Amazon S3. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.

To help categorize and manage your volumes and snapshots, you can tag them with metadata of your choice. For more information, see Tagging Your Amazon EC2 Resources (p. 610).

Amazon EBS Volume Types

Amazon EBS provides the following volume types, which differ in performance characteristics and price, so that you can tailor your storage performance and cost to the needs of your applications:

- General Purpose (SSD) Volumes (p. 524)
- Provisioned IOPS (SSD) Volumes (p. 525)
- Magnetic Volumes (p. 525)

The following table describes basic use cases and performance characteristics for each volume type.

Characteristic	General Purpose (SSD)	Provisioned IOPS (SSD)	Magnetic
Use cases	- System boot volumes - Virtual desktops - Small to medium sized databases - Development and test environments	- Critical business applications that require sustained IOPS performance above 3,000 IOPS - Large database workloads, such as: - MongoDB - Microsoft SQL Server - MySQL - PostgreSQL - Oracle	- Cold workloads where data is infrequently accessed - Scenarios where the lowest storage cost is important
Volume size	1 GiB - 1 TiB	4 GiB - 1 TiB	1 GiB - 1 TiB
Maximum throughput	128 MiB/s	128 MiB/s	40-90 MiB/s
IOPS performance	Has the ability to burst to 3,000 IOPS maximum, with a base performance of 3 IOPS/GiB	Consistently performs at provisioned level, up to 4,000 IOPS maximum	Averages 100 IOPS, with the ability to burst to hundreds of IOPS

Characteristic	General Purpose (SSD)	Provisioned IOPS (SSD)	Magnetic
API and CLI volume name	`gp2`	`io1`	`standard`

There are several factors that can affect the performance of Amazon EBS volumes, such as instance configuration, I/O characteristics, and workload demand. For more information about getting the most out of your Amazon EBS volumes, see Amazon EBS Volume Performance (p. 567).

For detailed pricing information about these volume types, see Amazon EBS Pricing.

General Purpose (SSD) Volumes

General Purpose (SSD) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies, the ability to burst to 3,000 IOPS for extended periods of time, and a base performance of 3 IOPS/GiB. General Purpose (SSD) volumes can range in size from 1 GiB to 1 TiB.

> **Note**
> Some AWS accounts created before 2012 might have access to Availability Zones in us-east-1, us-west-1, or ap-northeast-1 that do not support SSD volumes such as Provisioned IOPS (SSD) and General Purpose (SSD). If you are unable to create an SSD volume (or launch an instance with an SSD volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports General Purpose (SSD) and Provisioned IOPS (SSD) volumes by creating a 1 GiB General Purpose (SSD) volume in that zone.

IO Credits and Burst Performance

General Purpose (SSD) volume performance is governed by volume size, which dictates the base per-formance level of the volume and how quickly it accumulates I/O credits; larger volumes have higher base performance levels and accumulate I/O credits faster. I/O credits represent the available bandwidth that your General Purpose (SSD) volume can use to burst large amounts of I/O when more than the base performance is needed. The more credits your volume has for I/O, the more time it can burst beyond its base performance level and the better it performs when more performance is needed.

Each volume receives an initial I/O credit balance of 5,400,000 I/O credits, which is enough to sustain the maximum burst performance of 3,000 IOPS for 30 minutes. This initial credit balance is designed to provide a fast initial boot cycle for boot volumes and to provide a good bootstrapping experience for other applications. Volumes earn I/O credits every second at a base performance rate of 3 IOPS per GiB of volume size. For example, a 100 GiB General Purpose (SSD) volume has a base performance of 300 IOPS.

When your volume requires more than the base performance I/O level, it simply uses I/O credits in the credit balance to burst to the required performance level, up to a maximum of 3,000 IOPS. Volumes larger than 1,000 GiB have a base performance that is equal or greater than the maximum burst performance, and their I/O credit balance never depletes. When your volume uses fewer I/O credits than it earns in a second, unused I/O credits are added to the I/O credit balance. The maximum I/O credit balance for a volume is equal to the initial credit balance (5,400,000 I/O credits).

If your volume uses all of its I/O credit balance, the maximum performance of the volume will remain at the base performance level (the rate at which your volume earns credits) until I/O demand drops below the base level and unused credits are added to the I/O credit balance. The larger a volume is, the greater the base performance is and the faster it replenishes the credit balance.

The table below lists several volume sizes and the associated base performance of the volume (which is also the rate at which it accumulates I/O credits), the burst duration at the 3,000 IOPS maximum (when

starting with a full credit balance), and the time in seconds that the volume would take to refill an empty credit balance.

Volume size (GiB)	Base performance (IOPS)	Maximum burst duration @ 3,000 IOPS (seconds)	Seconds to fill empty credit balance
1	3	1,802	1,800,000
100	300	2,000	18,000
250	750	2,400	7,200
500	1,500	3,600	3,600
750	2,250	7,200	2,400
1,000	3,000	N/A*	N/A*

The burst duration of a volume is dependent on the size of the volume, the burst IOPS required, and the credit balance when the burst begins. This is shown in the equation below:

```
                        (Credit balance)
Burst duration  =  ---------------------------------
                   (Burst IOPS) - 3(Volume size in GiB)
```

If you notice that your volume performance is frequently limited to the base level (due to and empty I/O credit balance), you should consider using a larger General Purpose (SSD) volume (with a higher base performance level) or switching to a Provisioned IOPS (SSD) volume for workloads that require sustained IOPS performance greater than 3,000 IOPS.

Provisioned IOPS (SSD) Volumes

Provisioned IOPS (SSD) volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads, that are sensitive to storage performance and consistency in random access I/O throughput. You specify an IOPS rate when you create the volume, and Amazon EBS delivers within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year.

Note
Some AWS accounts created before 2012 might have access to Availability Zones in us-east-1, us-west-1, or ap-northeast-1 that do not support SSD volumes such as Provisioned IOPS (SSD) and General Purpose (SSD). If you are unable to create an SSD volume (or launch an instance with an SSD volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports General Purpose (SSD) and Provisioned IOPS (SSD) volumes by creating a 1 GiB General Purpose (SSD) volume in that zone.

A Provisioned IOPS (SSD) volume can range in size from 4 GiB to 1 TiB and you can provision up to 4,000 IOPS per volume. The ratio of IOPS provisioned to the volume size requested can be a maximum of 30; for example, a volume with 3,000 IOPS must be at least 100 GiB. You can stripe multiple volumes together in a RAID configuration for larger size and greater performance.

Magnetic Volumes

Magnetic volumes provide the lowest cost per gigabyte of all Amazon EBS volume types. Magnetic volumes are backed by magnetic drives and are ideal for workloads performing sequential reads, workloads where data is accessed infrequently, and scenarios where the lowest storage cost is important. These volumes deliver approximately 100 IOPS on average, with burst capability of up to hundreds of IOPS,

and they can range in size from 1 GiB to 1 TiB. Magnetic volumes can be striped together in a RAID configuration for larger size and greater performance.

If you need a greater number of IOPS or higher performance than Magnetic volume can provide, we recommend that you consider General Purpose (SSD) or Provisioned IOPS (SSD) volumes.

Creating an Amazon EBS Volume

You can create a new Amazon EBS volume that you can then attach to any Amazon EC2 instance within the same Availability Zone. You can choose to create an encrypted Amazon EBS volume, but encrypted volumes can only be attached to selected instance types. For more information, see Supported Instance Types (p. 565).

You can also create and attach Amazon EBS volumes when you launch instances by specifying a block device mapping. For more information, see Launching an Instance (p. 290) and Block Device Mapping (p. 591). You can restore volumes from previously created snapshots. For more information, see Restoring an Amazon EBS Volume from a Snapshot (p. 527).

If you are creating a volume for a high-performance storage scenario, you should make sure to use a Provisioned IOPS (SSD) volume and attach it to an instance with enough bandwidth to support your application, such as an EBS-optimized instance or an instance with 10 Gigabit network connectivity. For more information, see Amazon EC2 Instance Configuration (p. 568).

When a block of data on a newly created Amazon EBS volume is written to for the first time, you might experience longer than normal latency. To avoid the possibility of an increased write latency on a production workload, you should first write to all blocks on the volume to ensure optimal performance; this practice is called pre-warming the volume. For more information, see Pre-Warming Amazon EBS Volumes (p. 570).

To create a new Amazon EBS volume using the console

1. Open the Amazon EC2 console.
2. From the navigation bar, select the region in which you would like to create your volume. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see Resource Locations (p. 605).
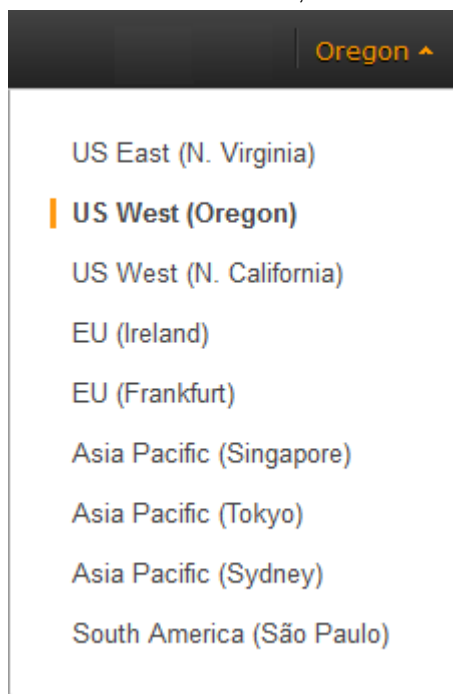
3. Click **Volumes** in the navigation pane.

4. Above the upper pane, click **Create Volume**.

5. In the **Create Volume** dialog box, in the **Volume Type** list, select **General Purpose (SSD)**, **Provisioned IOPS (SSD)** or **Magnetic**. For more information, see Amazon EBS Volume Types (p. 523).

   > **Note**
   > Some AWS accounts created before 2012 might have access to Availability Zones in us-east-1, us-west-1, or ap-northeast-1 that do not support SSD volumes such as Provisioned IOPS (SSD) and General Purpose (SSD). If you are unable to create an SSD volume (or launch an instance with an SSD volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports General Purpose (SSD) and Provisioned IOPS (SSD) volumes by creating a 1 GiB General Purpose (SSD) volume in that zone.

6. In the **Size** box, enter the size of the volume, in GiB.

7. For Provisioned IOPS (SSD) volumes, in the **IOPS** box, enter the maximum number of input/output operations per second (IOPS) that the volume should support.

8. In the **Availability Zone** list, select the Availability Zone in which to create the volume.

9. (Optional) To create an encrypted volume, select the **Encrypted** box and choose the master key you want to use when encrypting the volume. You can choose the default master key for your account, or you can choose any Customer Master Key (CMK) that you have previously created using the AWS Key Management Service. Available keys are visible in the **Master Key** drop down menu, or you can paste the full ARN of any key that you have access to. For more information, see the AWS Key Management Service Developer Guide.

   > **Note**
   > Encrypted volumes can only be attached to selected instance types. For more information, see Supported Instance Types (p. 565).

10. Click **Yes, Create**.

To create a new Amazon EBS volume using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- create-volume (AWS CLI)
- ec2-create-volume (Amazon EC2 CLI)
- New-EC2Volume (AWS Tools for Windows PowerShell)

Restoring an Amazon EBS Volume from a Snapshot

You can restore an Amazon EBS volume with data from a snapshot stored in Amazon S3. You need to know the ID of the snapshot you wish to restore your volume from and you need to have access permissions for the snapshot. For more information on snapshots, see Amazon EBS Snapshots (p. 559).

New volumes created from existing Amazon S3 snapshots load lazily in the background. This means that after a volume is created from a snapshot, there is no need to wait for all of the data to transfer from Amazon S3 to your Amazon EBS volume before your attached instance can start accessing the volume and all its data. If your instance accesses data that hasn't yet been loaded, the volume immediately downloads the requested data from Amazon S3, and continues loading the rest of the data in the background.

Amazon EBS volumes that are restored from encrypted snapshots are automatically encrypted. Encrypted volumes can only be attached to selected instance types. For more information, see Supported Instance Types (p. 565).

When a block of data on a newly restored Amazon EBS volume is accessed for the first time, you might experience longer than normal latency. To avoid the possibility of increased read or write latency on a production workload, you should first access all of the blocks on the volume to ensure optimal performance; this practice is called pre-warming the volume. For more information, see Pre-Warming Amazon EBS Volumes (p. 570).

To restore an Amazon EBS volume from a snapshot using the console

You can restore your Amazon EBS volume from a snapshot using the AWS Management Console as follows.

1. Open the Amazon EC2 console.
2. From the navigation bar, select the region that your snapshot is located in. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see Resource Locations (p. 605). If you need to restore the snapshot to a volume in a different region, you can copy your snapshot to the new region and then restore it to a volume in that region. For more information, see Copying an Amazon EBS Snapshot (p. 561).

3. Click **Volumes** in the navigation pane.
4. Click **Create Volume**.
5. In the **Create Volume** dialog box, in the **Volume Type** list, select **General Purpose (SSD)**, **Provisioned IOPS (SSD)** or **Magnetic**. For more information, see Amazon EBS Volume Types (p. 523).

   > **Note**
   > Some AWS accounts created before 2012 might have access to Availability Zones in us-east-1, us-west-1, or ap-northeast-1 that do not support SSD volumes such as Provisioned IOPS (SSD) and General Purpose (SSD). If you are unable to create an SSD volume (or launch an instance with an SSD volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports General Purpose (SSD) and Provisioned IOPS (SSD) volumes by creating a 1 GiB General Purpose (SSD) volume in that zone.

6. In the **Snapshot** field, start typing the ID or description of the snapshot from which you are restoring the volume, and select it from the list of suggested options.

Note

Volumes that are restored from encrypted snapshots can only be attached to instances that support Amazon EBS encryption. For more information, see Supported Instance Types (p. 565).

7. In the **Size** box, enter the size of the volume in GiB, or verify the that the default size of the snapshot is adequate.

Note

If you specify both a volume size and a snapshot ID, the size must be equal to or greater than the snapshot size. When you select a volume type and a snapshot ID, minimum and maximum sizes for the volume are shown next to the **Size** list. Any AWS Marketplace product codes from the snapshot are propagated to the volume.

8. For Provisioned IOPS (SSD) volumes, in the **IOPS** box, enter the maximum number of input/output operations per second (IOPS) that the volume can support.

9. In the **Availability Zone** list, select the Availability Zone in which to create the volume.

10. Click **Yes, Create**.

Important

If you restored a snapshot to a larger volume than the default for that snapshot, you need to extend the file system on the volume to take advantage of the extra space. For more information, see Expanding the Storage Space of a Volume (p. 545).

To restore a new Amazon EBS volume using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- create-volume (AWS CLI)
- ec2-create-volume (Amazon EC2 CLI)
- New-EC2Volume (AWS Tools for Windows PowerShell)

Attaching an Amazon EBS Volume to an Instance

You can attach your Amazon EBS volume to one of your instances in the same Availability Zone.

To attach an Amazon EBS volume to an instance using the console

1. Open the Amazon EC2 console.
2. Click **Volumes** in the navigation pane.
3. Select a volume and then click **Attach Volume**.
4. In the **Attach Volume** dialog box, start typing the name or ID of the instance to attach the volume to in the **Instance** box, and select it from the list of suggestion options (only instances in the same Availability Zone as the volume are displayed). Encrypted volumes can only be attached to instances that support Amazon EBS encryption. For more information, see Supported Instance Types (p. 565).
5. Verify that the suggested device name is suitable, or enter a device name in the **Device** box. For more information about naming conventions and restrictions, see Device Naming (p. 530).
6. Click **Attach** to attach the volume to the instance. The volume and instance must be in the same Availability Zone.

Note

Volumes that have been mounted using the **mount** command are lost after restarting. To mount this Amazon EBS volume on every system reboot, add an entry for the device to the `/etc/fstab` file. For more information, see Making an Amazon EBS Volume Available for Use (p. 531).

If a volume has an AWS Marketplace product code:

- The volume can only be attached to the root device of a stopped instance.
- You must be subscribed to the AWS Marketplace code that is on the volume.
- The configuration (instance type, operating system) of the instance must support that specific AWS Marketplace code. For example, you cannot take a volume from a Windows instance and attach it to a Linux instance.
- AWS Marketplace product codes are copied from the volume to the instance.

To attach an Amazon EBS volume to an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- attach-volume (AWS CLI)
- ec2-attach-volume (Amazon EC2 CLI)
- Add-EC2Volume (AWS Tools for Windows PowerShell)

Device Naming

The following table lists the available device names on Amazon EC2. You can specify these names when attaching a volume to a running instance or when launching an instance using a block device mapping. The block device driver for the instance assigns the actual volume names when mounting the volumes, and these names can be different than the names that Amazon EC2 recommends. For more information about instance store volumes, see Amazon EC2 Instance Store (p. 581). For information about the root device storage, see Amazon EC2 Root Device Volume (p. 14).

Virtualization Type	Available	Reserved for Root	Used for Instance Store Volumes	Recommended for EBS Volumes
Paravirtual	/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]
HVM	/dev/sd[a-z] /dev/xvd[b-c][a-z]	Differs by AMI /dev/sda1 or /dev/xvda	/dev/sd[b-e] /dev/sd[b-y] (hs1.8xlarge)	/dev/sd[f-p]

> **Warning**
> Although you can attach your Amazon EBS volumes using the device names used to attach instance store volumes, we strongly recommend that you don't because the behavior can be unpredictable.

Depending on the block device driver of the kernel, the device might be attached with a different name than what you specify. For example, if you specify a device name of `/dev/sdh`, your device might be renamed `/dev/xvdh` or `/dev/hdh` by the kernel; in most cases, the trailing letter remains the same. In some versions of Red Hat Enterprise Linux (and its variants, such as CentOS), even the trailing letter might also change (where `/dev/sda` could become `/dev/xvde`). In these cases, each device name trailing letter is incremented the same number of times. For example, `/dev/sdb` would become `/dev/xvdf`

and `/dev/sdc` would become `/dev/xvdg`. Amazon Linux AMIs create a symbolic link from the renamed device path to the name you specify, but other AMIs might behave differently.

You cannot attach volumes that share the same device letters both with and without trailing digits. For example, if you attach a volume as `/dev/sdc` and another volume as `/dev/sdc1`, only `/dev/sdc` is visible to the instance. To use trailing digits in device names, you must use trailing digits on all device names that share the same base letters (such as `/dev/sdc1`, `/dev/sdc2`, `/dev/sdc3`).

Hardware virtual machine (HVM) AMIs don't support the use of trailing numbers on device names.

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]`.

Volume Limits

Although Amazon EC2 does not impose limitations on how many volumes you can attach to an instance, there are several factors you need to consider when attaching multiple volumes. The upper limit of volumes you should attach to your instance is dependent on your operating system (Linux instances can reliably handle more volumes than Windows instances). Another consideration is whether you need increased I/O bandwidth or increased storage capacity.

Bandwidth vs Capacity

For consistent and predictable bandwidth use cases, use EBS-optimized or 10 Gigabit network connectivity instances and General Purpose (SSD) or Provisioned IOPS (SSD) volumes. Follow the guidance in the Amazon EC2 Instance Configuration (p. 568) topic to match the IOPS you have provisioned for your volumes to the bandwidth available from your instances for maximum performance. For RAID configurations, many administrators find that arrays larger than 8 volumes have diminished performance returns due to increased I/O overhead; test your individual application performance and tune it as required.

For increased capacity use cases, you can attach many more volumes, but attaching more than 40 volumes (this quantity is only possible on Linux instances) might result in boot failures. If you experience boot problems on an instance with many volumes attached, you can stop the instance, detach any volumes that are non-essential to the boot process, and then reattach them after the instance is running.

Making an Amazon EBS Volume Available for Use

After you attach an Amazon EBS volume to your instance, it is exposed as a block device. You can format the volume with any file system and then mount it. After you make the Amazon EBS volume available for use, you can access it in the same ways that you access any other volume. Any data written to this file system is written to the Amazon EBS volume and is transparent to applications using the device.

Note that you can take snapshots of your Amazon EBS volume for backup purposes or to use as a baseline when you create another volume. For more information, see Amazon EBS Snapshots (p. 559).

Making the Volume Available on Linux

Use the following procedure to make the volume available. Note that you can get directions for volumes on a Windows instance from Making the Volume Available on Windows in the *Amazon EC2 User Guide for Microsoft Windows Instances*.

To make an Amazon EBS volume available for use on Linux

1. Connect to your instance using SSH. For more information, see Connect to Your Instance (p. 28).
2. Depending on the block device driver of the kernel, the device might be attached with a different name than what you specify. For example, if you specify a device name of `/dev/sdh`, your device might be renamed `/dev/xvdh` or `/dev/hdh` by the kernel; in most cases, the trailing letter remains

the same. In some versions of Red Hat Enterprise Linux (and its variants, such as CentOS), even the trailing letter might also change (where `/dev/sda` could become `/dev/xvde`). In these cases, each device name trailing letter is incremented the same number of times. For example, `/dev/sdb` would become `/dev/xvdf` and `/dev/sdc` would become `/dev/xvdg`. Amazon Linux AMIs create a symbolic link from the renamed device path to the name you specify, but other AMIs might behave differently.

Use the **lsblk** command to view your available disk devices and their mount points (if applicable) to help you determine the correct device name to use.

```
[ec2-user ~]$ lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvdf  202:80   0  100G  0 disk
xvda1 202:1    0    8G  0 disk /
```

The output of **lsblk** removes the `/dev/` prefix from full device paths. In this example, `/dev/xvda1` is mounted as the root device (note the `MOUNTPOINT` is listed as `/`, the root of the Linux file system hierarchy), and `/dev/xvdf` is attached, but it has not been mounted yet.

3. Determine whether you need to create a file system on the volume. New volumes are raw block devices, and you need to create a file system on them before you can mount and use them. Volumes that have been restored from snapshots likely have a file system on them already; if you create a new file system on top of an existing file system, the operation overwrites your data. Use the **sudo file -s _device_** command to list special information, such as file system type.

```
[ec2-user ~]$ sudo file -s /dev/xvdf
/dev/xvdf: data
```

If the output of the previous command shows simply `data` for the device, then there is no file system on the device and you need to create one. You can go on to Step 4 (p. 532). If you run this command on a device that contains a file system, then your output will be different.

```
[ec2-user ~]$ sudo file -s /dev/xvda1
/dev/xvda1: Linux rev 1.0 ext4 filesystem data, UUID=1701d228-e1bd-4094-
a14c-8c64d6819362 (needs journal recovery) (extents) (large files) (huge
files)
```

In the previous example, the device contains `Linux rev 1.0 ext4 filesystem data`, so this volume does not need a file system created (you can skip Step 4 (p. 532) if your output shows file system data).

4. (Conditional) Use the following command to create an ext4 file system on the volume. Substitute the device name (such as `/dev/xvdf`) for _device_name_. Depending on the requirements of your application or the limitations of your operating system, you can choose a different file system type, such as ext3 or XFS.

    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 volume that was restored from a snapshot), 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.

```
[ec2-user ~]$ sudo mkfs -t ext4 device_name
```

5. Use the following command to create a mount point directory for the volume. The mount point is where the volume is located in the file system tree and where you read and write files to after you mount the volume. Substitute a location for *mount_point*, such as /data.

```
[ec2-user ~]$ sudo mkdir mount_point
```

6. Use the following command to mount the volume at the location you just created.

```
[ec2-user ~]$ sudo mount device_name mount_point
```

7. (Optional) To mount this Amazon EBS volume on every system reboot, add an entry for the device to the /etc/fstab file.

    a. Create a backup of your /etc/fstab file that you can use if you accidentally destroy or delete this file while you are editing it.

    ```
    [ec2-user ~]$ sudo cp /etc/fstab /etc/fstab.orig
    ```

    b. Open the /etc/fstab file using your favorite text editor, such as **nano** or **vim**.

    c. Add a new line to the end of the file for your volume using the following format.

    ```
    device_name  mount_point  file_system_type  fs_mntops  fs_freq  fs_passno
    ```

    The last three fields on this line are the file system mount options, the dump frequency of the file system, and the order of file system checks done at boot time. If you don't know what these values should be, then use the values in the example below for them (defaults,nofail 0 2). For more information on /etc/fstab entries, see the **fstab** manual page (by entering **man fstab** on the command line). For example, to mount the ext4 file system on the device /dev/xvdf at the mount point /data, add the following entry to /etc/fstab.

    > **Note**
    > If you ever intend to boot your instance without this volume attached (for example, so this volume could move back and forth between different instances), you should add the nofail mount option that allows the instance to boot even if there are errors in mounting the volume. Debian derivatives, such as Ubuntu, must also add the noboot-wait mount option.

    ```
    /dev/xvdf       /data   ext4    defaults,nofail     0       2
    ```

    d. After you've added the new entry to /etc/fstab, you need to check that your entry works. Run the **sudo mount -a** command to mount all file systems in /etc/fstab.

    ```
    [ec2-user ~]$ sudo mount -a
    ```

    If the previous command does not produce an error, then your /etc/fstab file is OK and your file system will mount automatically at the next boot. If the command does produce any errors, examine the errors and try to correct your /etc/fstab.

    > **Warning**
    > Errors in the /etc/fstab file can render a system unbootable. Do not shut down a system that has errors in the /etc/fstab file.

e. (Optional) If you are unsure how to correct `/etc/fstab` errors, you can always restore your
backup `/etc/fstab` file with the following command.

```
[ec2-user ~]$ sudo mv /etc/fstab.orig /etc/fstab
```

Viewing Volume Information

You can view descriptive information for all of your volumes in a selected region at a time in the AWS
Management Console. You can also view detailed information about a single volume, including the size,
volume type, whether or not the volume is encrypted, which master key was used to encrypt the volume,
and the specific instance to which the volume is attached.

To view information about an Amazon EBS volume using the console

1. Open the Amazon EC2 console.
2. Click **Volumes** in the navigation pane.
3. To view more information about a volume, select it.

To view information about an Amazon EBS volume using the command line

You can use one of the following commands. For more information about these command line interfaces,
see Accessing Amazon EC2 (p. 3).

- describe-volumes (AWS CLI)
- ec2-describe-volumes (Amazon EC2 CLI)
- Get-EC2Volume (AWS Tools for Windows PowerShell)

Monitoring the Status of Your Volumes

Amazon Web Services (AWS) automatically provides data, such as Amazon CloudWatch metrics and
volume status checks, that you can use to monitor your Amazon Elastic Block Store (Amazon EBS)
volumes.

Contents

Monitoring Volumes with CloudWatch

CloudWatch metrics are statistical data that you can use to view, analyze, and set alarms on the opera-
tional behavior of your volumes.

The following table describes the types of monitoring data available for your Amazon EBS volumes.

Type	Description
Basic	Data is available automatically in 5-minute periods at no charge. This includes data for the root device volumes for Amazon EBS-backed instances.
Detailed	Provisioned IOPS (SSD) volumes automatically send one-minute metrics to CloudWatch.

When you get data from CloudWatch, you can include a `Period` request parameter to specify the granularity of the returned data. This is different than the period that we use when we collect the data (5-minute periods). We recommend that you specify a period in your request that is equal to or larger than the collection period to ensure that the returned data is valid.

You can get the data using either the Amazon CloudWatch API or the Amazon EC2 console. The console takes the raw data from the Amazon CloudWatch API and displays a series of graphs based on the data. Depending on your needs, you might prefer to use either the data from the API or the graphs in the console.

Amazon EBS Metrics

You can use the Amazon CloudWatch `GetMetricStatistics` API to get any of the Amazon EBS volume metrics listed in the following table. Similar metrics are grouped together in the table, and the metrics in the first two rows are also available for the local stores on Amazon EC2 instances.

Metric	Description

`VolumeReadBytes`
 	
`VolumeWriteBytes`
	The total number of bytes transferred in a specified period of time. Data is only reported to Amazon CloudWatch when the volume is active. If the volume is idle, no data is reported to Amazon CloudWatch.
	
	Units: Bytes

`VolumeReadOps`
 	
`VolumeWriteOps`
	The total number of I/O operations in a specified period of time.
	
	Note
	To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period.
	
	Units: Count

`VolumeTotalReadTime`
 	
`VolumeTotalWriteTime`
	The total number of seconds spent by all operations that completed in a specified period of time. If multiple requests are submitted at the same time, this total could be greater than the length of the period. For example, for a period of 5 minutes (300 seconds): if 700 operations completed during that period, and each operation took 1 second, the value would be 700 seconds.
	
	Units: Seconds

`VolumeIdleTime`	The total number of seconds in a specified period of time when no read or write operations were submitted. Units: Seconds
`VolumeQueueLength`	The number of read and write operation requests waiting to be completed in a specified period of time. Units: Count

Metric	Description
VolumeThroughputPercentage	Used with Provisioned IOPS (SSD) volumes only. The percentage of I/O operations per second (IOPS) delivered of the total IOPS provisioned for an Amazon EBS volume. Provisioned IOPS (SSD) volumes deliver within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year.
	Note
	During a write, if there are no other pending I/O requests in a minute, the metric value will be 100 percent. Also, a volume's I/O performance may become degraded temporarily due to an action you have taken (e.g., creating a snapshot of a volume during peak usage, running the volume on a non-EBS-optimized instance, accessing data on the volume for the first time).
	Units: Percent
VolumeConsumedReadWriteOps	Used with Provisioned IOPS (SSD) volumes only. The total amount of read and write operations (normalized to 256K capacity units) consumed in a specified period of time.
	I/O operations that are smaller than 256K each count as 1 consumed IOPS. I/O operations that are larger than 256K are counted in 256K capacity units. For example, a 1024K I/O would count as 4 consumed IOPS.
	Units: Count

Graphs in the Amazon EC2 console

After you create a volume, you can go to the Amazon EC2 console and view the volume's monitoring graphs. They're displayed when you select the volume on the **Volumes** page in the EC2 console. A **Monitoring** tab is displayed next to the volume's **Description** tab. The following table lists the graphs that are displayed. The column on the right describes how the raw data metrics from the Amazon CloudWatch API are used to produce each graph. The period for all the graphs is 5 minutes.

Graph Name	Description Using Raw Metrics
Read Bandwidth (KiB/s)	Sum(VolumeReadBytes) / Period / 1024
Write Bandwidth (KiB/s)	Sum(VolumeWriteBytes) / Period / 1024
Read Throughput (Ops/s)	Sum(VolumeReadOps) / Period
Write Throughput (Ops/s)	Sum(VolumeWriteOps) / Period
Avg Queue Length (ops)	Avg(VolumeQueueLength)
% Time Spent Idle	Sum(VolumeIdleTime) / 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 Amazon EBS volumes are impaired, and to help you control how a potentially inconsistent volume is handled.

Volume status checks are automated tests that return a pass or fail status. If all checks pass, the status of the volume is `ok`. If a check fails, the status of the volume is `impaired`. If the status is `insufficient-data`, the checks may still be in progress on the volume. You can view the results of volume status checks to identify any impaired volumes and take any necessary actions.

When Amazon EBS determines that a volume's data is potentially inconsistent, the default is that it disables I/O to the volume from any attached EC2 instances, which helps to prevent data corruption. After I/O is disabled, the next volume status check fails, and the volume status is `impaired`. In addition, you'll see an event that lets you know that I/O is disabled, and that you can resolve the impaired status of the volume by enabling I/O to the volume. We wait until you enable I/O to give you the opportunity to decide whether to continue to let your instances use the volume, or to run a consistency check using a command, such as **fsck**, 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 I/O performance status check compares actual volume performance to the expected performance of a volume and alerts you if the volume is performing below expectations. This status check is only available for Provisioned IOPS (SSD) volumes that are attached to an instance and is not valid for General Purpose (SSD) and Magnetic volumes. The I/O performance status check is performed once every minute and CloudWatch collects this data every 5 minutes, so it may take up to 5 minutes from the moment you attach a Provisioned IOPS (SSD) volume to an instance for this check to report the I/O performance status.

The following table lists statuses for Amazon EBS volumes.

Overall Volume Status	I/O Enabled Status	I/O Performance Status (Provisioned IOPS (SSD) volumes only)
ok	Enabled (I/O Enabled or I/O Auto-Enabled)	Normal (Volume performance is as expected)

Overall Volume Status	I/O Enabled Status	I/O Performance Status (Provisioned IOPS (SSD) volumes only)
`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 Amazon EC2 console, the API, or the command line interface.

To view status checks in the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Volumes**.
3. On the **EBS Volumes** page, the **Volume Status** column lists the operational status of each volume.
4. To view an individual volume's status, select the volume, and then click the **Status Checks** tab.

5. If you have a volume with a failed status check (status is `impaired`), see Working with an Impaired Volume (p. 540).

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. 539).

To view volume status information with the command line

You can use one of the following commands to view the status of your Amazon EBS volumes. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-volume-status (AWS CLI)
- ec2-describe-volume-status (Amazon EC2 CLI)
- Get-EC2VolumeStatus (AWS Tools for Windows PowerShell)

Monitoring Volume Events

When Amazon EBS determines that a volume's data is potentially inconsistent, it disables I/O to the volume from any attached EC2 instances by default. This causes the volume status check to fail, and creates a volume status event that indicates the cause of the failure.

To automatically enable I/O on a volume with potential data inconsistencies, change the setting of the `AutoEnableIO` volume attribute. For more information about changing this attribute, see Working with an Impaired Volume (p. 540).

Each event includes a start time that indicates the time at which the event occurred, and a duration that indicates how long I/O for the volume was disabled. The end time is added to the event when I/O for the volume is enabled.

Volume status events include one of the following descriptions:

Awaiting Action: Enable IO
    Volume data is potentially inconsistent. I/O is disabled for the volume until you explicitly enable it. The event description changes to **IO Enabled** after you explicitly enable I/O.
IO Enabled
    I/O operations were explicitly enabled for this volume.
IO Auto-Enabled
    I/O operations were automatically enabled on this volume after an event occurred. We recommend that you check for data inconsistencies before continuing to use the data.
Normal
    For Provisioned IOPS (SSD) volumes only. Volume performance is as expected.
Degraded
    For Provisioned IOPS (SSD) volumes only. Volume performance is below expectations.
Severely Degraded
    For Provisioned IOPS (SSD) volumes only. Volume performance is well below expectations.
Stalled
    For Provisioned IOPS (SSD) volumes only. Volume performance is severely impacted.

You can view events for your volumes using the Amazon EC2 console, the API, or the command line interface.

To view events for your volumes in the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Events**.
3. All instances and volumes that have events are listed. You can filter by volume to view only volume status. You can also filter on specific status types.
4. Select a volume to view its specific event.

If you have a volume where I/O is disabled, see Working with an Impaired Volume (p. 540). If you have a volume where I/O performance is below normal, this might be a temporary condition due to an action you have taken (e.g., creating a snapshot of a volume during peak usage, running the volume on an instance that cannot support the I/O bandwidth required, accessing data on the volume for the first time, etc.).

To view events for your volumes with the command line

You can use one of the following commands to view event information for your Amazon EBS volumes. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-volume-status (AWS CLI)
- ec2-describe-volume-status (Amazon EC2 CLI)
- Get-EC2VolumeStatus (AWS Tools for Windows PowerShell)

Working with an Impaired Volume

This section discusses your options if a volume is impaired because the volume's data is potentially inconsistent.

Options
- Option 1: Perform a Consistency Check on the Volume Attached to its Instance (p. 540)
- Option 2: Perform a Consistency Check on the Volume Using Another Instance (p. 541)
- Option 3: Delete the Volume If You No Longer Need It (p. 542)

Option 1: Perform a Consistency Check on the Volume Attached to its Instance

The simplest option is to enable I/O and then perform a data consistency check on the volume while the volume is still attached to its Amazon EC2 instance.

To perform a consistency check on an attached volume

1. Stop any applications from using the volume.

2. Enable I/O on the volume.

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

    b. In the navigation pane, click **Volumes**.

    c. Select the volume on which you want to enable I/O operations.

    d. In the details pane, click **Enable Volume IO**.

> Volumes: ▌ vol-d882c69b
>
> ⚠ IO operations have been disabled since 16 hours and 58 minutes ago ago. Data inconsistencies may exist. | **Troubleshoot** | **Enable Volume IO** |
>
> | **Description** | Status Checks | Monitoring | Tags |
> | --- | --- | --- | --- |
>
> | | | | |
> | --- | --- | --- | --- |
> | Volume ID | vol-d882c69b | Alarm status | *None* |
> | Capacity | 100 GiB | Snapshot | – |
> | Created | November 21, 2013 3:42:01 PM UTC+2 | Availability Zone | us-east-1d |
> | State | available | Attachment information | |
> | Volume type | io1 | IOPS | 500 |
> | Product codes | – | | |

    e. In **Enable Volume IO**, click **Yes, Enable**.

3. Check the data on the volume.

    a. Run the **fsck** command.

    b. (Optional) Review any available application or system logs for relevant error messages.

    c. If the volume has been impaired for more than 20 minutes you can contact support. Click **Troubleshoot**, and then on the **Troubleshoot Status Checks** dialog box, click **Contact Support** to submit a support case.

For information about using the command line interface to enable I/O for a volume, see ec2-enable-volume-io in the *Amazon EC2 Command Line Reference*. For information about using the API to enable I/O for a volume, see EnableVolumeIO in the *Amazon EC2 API Reference*.

Option 2: Perform a Consistency Check on the Volume Using Another Instance

Use the following procedure to check the volume outside your production environment.

> **Important**
> This procedure may cause the loss of write I/Os that were suspended when volume I/O was disabled.

To perform a consistency check on a volume in isolation

1. Stop any applications from using the volume.
2. Detach the volume from the instance.

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

    b. In the navigation pane, click **Volumes**.

    c. Select the volume that you want to detach.

    d. Click **Actions**, and then click **Force Detach Volume**. You'll be prompted for confirmation.

3. Enable I/O on the volume.

a. In the navigation pane, click **Volumes**.

b. Select the volume that you detached in the previous step.

c. In the details pane, click **Enable Volume IO**.

Volumes: | vol-d882c69b

⚠ IO operations have been disabled since 16 hours and 58 minutes ago ago. Data inconsistencies may exist. | Troubleshoot | Enable Volume IO |

| Description | Status Checks | Monitoring | Tags |

Volume ID	vol-d882c69b		Alarm status	None
Capacity	100 GiB		Snapshot	-
Created	November 21, 2013 3:42:01 PM UTC+2		Availability Zone	us-east-1d
State	available		Attachment information	
Volume type	io1		IOPS	500
Product codes	-			

d. In the **Enable Volume IO** dialog box, click **Yes, Enable**.

4. Attach the volume to another instance. For information, see Launch Your Instance (p. 289) and Attaching an Amazon EBS Volume to an Instance (p. 529).

5. Check the data on the volume.

a. Run the **fsck** command.

b. (Optional) Review any available application or system logs for relevant error messages.

c. If the volume has been impaired for more than 20 minutes, you can contact support. Click **Troubleshoot**, and then in the troubleshooting dialog box, click **Contact Support** to submit a support case.

For information about using the command line interface to enable I/O for a volume, see ec2-enable-volume-io in the *Amazon EC2 Command Line Reference*. For information about using the API to enable I/O for a volume, see EnableVolumeIO in the *Amazon EC2 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. 545).

If you have a recent snapshot that backs up the data on the volume, you can create a new volume from the snapshot. For information about creating a volume from a snapshot, see Restoring an Amazon EBS Volume from a Snapshot (p. 527).

Working with the AutoEnableIO Volume Attribute

When Amazon EBS determines that a volume's data is potentially inconsistent, it disables I/O to the volume from any attached EC2 instances by default. This causes the volume status check to fail, and creates a volume status event that indicates the cause of the failure. If the consistency of a particular volume is not a concern, and you prefer that the volume be made available immediately if it's impaired, you can override the default behavior by configuring the volume to automatically enable I/O. If you enable the AutoEnableIO volume attribute, I/O between the volume and the instance is automatically reenabled and the volume's status check will pass. In addition, you'll see an event that lets you know that the volume was in a potentially inconsistent state, but that its I/O was automatically enabled. When this event occurs, you should check the volume's consistency and replace it if necessary. For more information, see Monitoring Volume Events (p. 539).

This section explains how to view and modify the `AutoEnableIO` attribute of a volume using the Amazon EC2 console, the command line interface, or the API.

To view the AutoEnableIO attribute of a volume in the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Volumes**.
3. Select the volume.
4. In the lower pane, click the **Status Checks** tab.
5. In the **Status Checks** tab, **Auto-Enable IO** displays the current setting for your volume, either `Enabled` or `Disabled`.
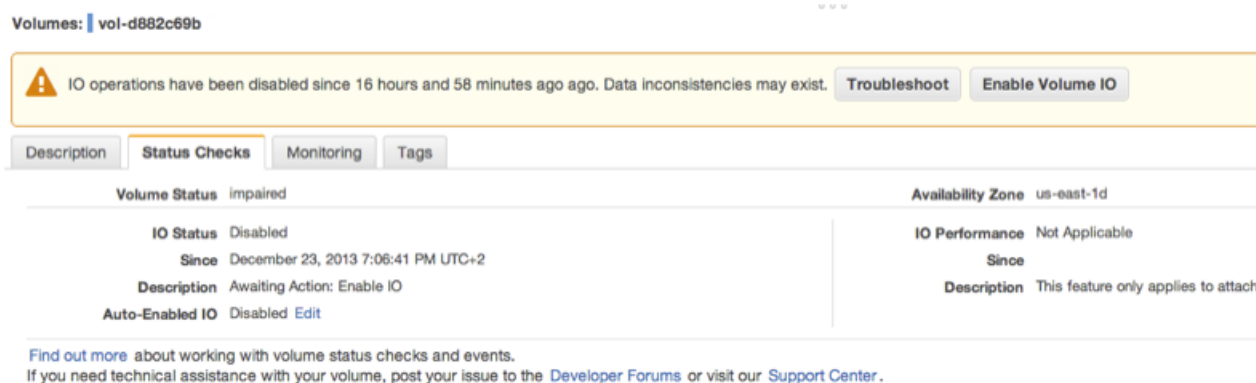
To modify the AutoEnableIO attribute of a volume in the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Volumes**.
3. Select the volume.
4. At the top of the **Volumes** page, click **Actions**.
5. Click **Change Auto-Enable IO Setting**.
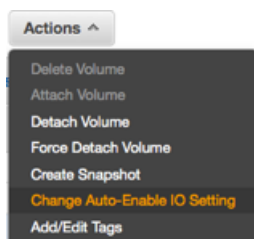
6. In the **Change Auto-Enable IO Setting** dialog box, select the **Auto-Enable Volume IO** option to automatically enable I/O for an impaired volume. To disable the feature, clear the option.
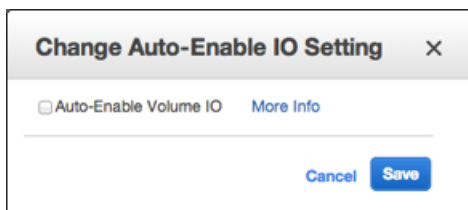
7.   Click **Save**.

Alternatively, instead of completing steps 4-6 in the previous procedure, go to the **Status Checks** tab and click **Edit**.

To view or modify the AutoEnableIO attribute of a volume with the command line

You can use one of the following commands to view the `AutoEnableIO` attribute of your Amazon EBS volumes. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-volume-attribute (AWS CLI)
- ec2-describe-volume-attribute (Amazon EC2 CLI)
- Get-EC2VolumeAttribute (AWS Tools for Windows PowerShell)

To modify the `AutoEnableIO` attribute of a volume, you can use one of the commands below.

- modify-volume-attribute (AWS CLI)
- ec2-modify-volume-attribute (Amazon EC2 CLI)
- Edit-EC2VolumeAttribute (AWS Tools for Windows PowerShell)

Detaching an Amazon EBS Volume from an Instance

You can detach an Amazon EBS volume from an instance explicitly or by terminating the instance. However, if the instance that the volume is attached to is running, you must unmount the volume (from the instance) before you detach it. Failure to do so results in the volume being stuck in the busy state while it is trying to detach, which could possibly damage the file system or the data it contains.

If an Amazon EBS volume is the root device of an instance, you must stop the instance before you can detach the volume.

When a root volume with an AWS Marketplace product code is detached from an instance, the product code is no longer associated with the instance.

> **Important**
> After you detach a volume, you are still charged for volume storage as long as the storage amount exceeds the limit of the Free Usage Tier. You must delete a volume to avoid incurring further charges. For more information, see Deleting an Amazon EBS Volume (p. 545).

This example unmounts the volume and then explicitly detaches it from the instance. This is useful when you want to terminate an instance or attach a volume to a different instance. To verify that the volume is no longer attached to the instance, see Viewing Volume Information (p. 534).

Note that you can reattach a volume that you detached (without unmounting it), but it might not get the same mount point and the data on the volume might be out of sync if there were writes to the volume in progress when it was detached.

To detach an Amazon EBS volume using the console

1.   First, use the following command to unmount the `/dev/sdh` device.

```
[ec2-user ~]$ umount -d /dev/sdh
```

2.   Open the Amazon EC2 console.
3.   Click **Volumes** in the navigation pane.

4. Select a volume and then click **Detach Volume**.
5. In the confirmation dialog box, click **Yes, Detach**.

To detach an Amazon EBS volume from an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

*   detach-volume (AWS CLI)
*   ec2-detach-volume (Amazon EC2 CLI)
*   Dismount-EC2Volume (AWS Tools for Windows PowerShell)

Troubleshooting

If your volume stays in the *detaching* state, you can force the detachment by clicking **Force Detach**. Forcing the detachment can lead to data loss or a corrupted file system. Use this option only as a last resort to detach a volume from a failed instance, or if you are detaching a volume with the intention of deleting it. The instance doesn't get an opportunity to flush file system caches or file system metadata. If you use this option, you must perform file system check and repair procedures.

If you've tried to force the volume to detach multiple times over several minutes and it stays in the *detaching* state, you can post a request for help to the Amazon EC2 forum. To help expedite a resolution, include the volume ID and describe the steps that you've already taken.

Deleting an Amazon EBS Volume

After you no longer need a volume, you can delete it. After deletion, its data is gone and the volume can't be attached to any instance. However, before deletion, you can store a snapshot of the volume, which you can use to recreate the volume later.

To delete a volume using the console

1. Open the Amazon EC2 console.
2. Click **Volumes** in the navigation pane.
3. Select a volume and click **Delete Volume**.
4. In the confirmation dialog box, click **Yes, Delete**.

To delete an Amazon EBS volume using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

*   delete-volume (AWS CLI)
*   ec2-delete-volume (Amazon EC2 CLI)
*   Remove-EC2Volume (AWS Tools for Windows PowerShell)

Expanding the Storage Space of a Volume

Sometimes, you might need to increase the storage space of an existing volume without losing the data that is on the volume. You can expand the storage space of an Amazon EBS volume by migrating your data to a larger volume and then extending the file system on the volume to recognize the newly-available space. After you verify that your new volume is working properly, you can delete the old volume.

Tasks

- Migrating Your Data to a Larger Volume (p. 546)
- Extending a Linux File System (p. 547)
- Deleting the Old Volume (p. 549)

If you need to expand the storage space of a volume on a Windows instance, see Expanding the Storage Space of a Volume in the *Amazon EC2 User Guide for Microsoft Windows Instances*.

Migrating Your Data to a Larger Volume

To migrate your data to a larger Amazon EBS volume

1. Open the Amazon EC2 console.

2. Ensure that the instance's **Shutdown Behavior** value is set to **Stop** and not **Terminate**. If it is already set to **Stop**, go on to step 3.

    a. In the navigation pane, click **Instances**, right-click on the instance to check, and then select **Change Shutdown Behavior**.

    b. If the **Shutdown behavior** is set to **Terminate**, select **Stop** from the list and click **Apply**.

        If the **Shutdown behavior** is set to **Stop**, click **Cancel**.
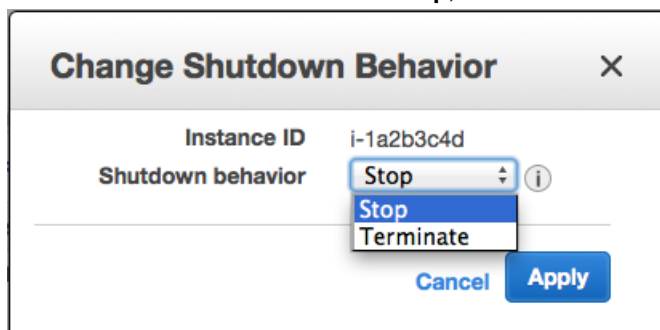
3. Stop the instance. For more information about how to stop an instance, see Stopping and Starting Your Instances (p. 310).

4. Create a snapshot of the volume to expand.

    a. In the navigation pane, click **Volumes**, right-click on the volume to be expanded, and select **Create Snapshot**.

    b. Enter a **Name** and **Description** for the snapshot, and click **Yes, Create**.

5. Create a new volume from the snapshot.

    a. In the navigation pane, click **Snapshots**.

    b. When the status of the snapshot that you just created is set to **completed**, select the snapshot and click **Create Volume**.

        Note
        It can take several minutes for the snapshot to complete.

    c. In the **Create Volume** dialog box, select the desired volume type, enter the new size for the volume, set the Availability Zone to the same Availability Zone as the instance, and click **Yes, Create**.

6. Detach the old volume.

    a. In the navigation pane, click **Volumes**, select the old volume from the list of volumes, and make note of the value of *device name* in **Attachment information**. The attachment information value takes the following form:

    ```
    instance information:device name
    ```

    b. Right-click the old volume and select **Detach Volume**.

    c. In the **Detach Volume** dialog box, click **Yes, Detach**. It may take several minutes for the volume to be detached.

7. Attach the newly expanded volume

    a. In the navigation pane, click **Volumes**, select the new volume from the list of volumes, right-click the new volume, and select **Attach Volume**.

    b. Start typing the name or ID of the instance in the **Instance** field, select the instance, enter the same device name retrieved in , and click **Yes, Attach**.

8. Restart the instance.

    a. In the navigation pane, click **Instances**, right-click the instance, and select **Start**.

    b. In the **Start Instances** dialog box, select **Yes, Start**. If the instance fails to start, and the volume being expanded is a root volume, verify that you attached the expanded volume using the same device name as the original volume (root volumes must be attached as `/dev/sda1`).

       Important
       Only EC2-VPC instances with Elastic IP addresses retain their public IP address when they are stopped. If your instance is running in EC2-Classic, the EIP address is disassociated when the instance is stopped, and you must re-associate the EIP after restarting the instance. For more information, see Elastic IP Addresses (EIP) (p. 489). If your instance is not using an EIP, then you need to retrieve your new public DNS name for your instance from the Instances page of the Amazon EC2 console to connect to it.

After the instance has started, you can check the file system size to see if your instance recognizes the larger volume space. On Linux, use the **df -h** command to check the file system size.

```
[ec2-user ~]$ 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
```

If the size does not reflect your newly-expanded volume, you must extend the file system your device so that your instance can use the new space. For more information, see Extending a Linux File System (p. 547).

Extending a Linux File System

In Linux, you use a file system-specific command to resize the file system to the larger size of the new volume. This command works even if the volume you wish to extend is the root volume. For `ext2`, `ext3`, and `ext4` file systems, this command is **resize2fs**. For XFS file systems, this command is **xfs_growfs**. For other file systems, refer to the specific documentation for those file systems for instructions on extending them.

If you are unsure of which file system you are using, you can use the **file -s** command to list the file system data for a device. The following example shows a Linux `ext4` file system and an SGI XFS file system.

```
[ec2-user ~]$ sudo file -s /dev/xvd*
/dev/xvda1: Linux rev 1.0 ext4 filesystem data ...
/dev/xvdf:  SGI XFS filesystem data ...
```

> **Note**
> If the volume you are extending has been partitioned, you need to increase the size of the partition before you can resize the file system. For more information, see Expanding a Linux Partition (p. 549).

To check if your volume partition needs resizing

- Use the **lsblk** command to list the block devices attached to your instance. The example below shows three volumes: `/dev/xvda`, `/dev/xvdb`, and `/dev/xvdf`.

```
[ec2-user ~]$ lsblk
NAME     MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda     202:0    0  30G  0 disk
  xvda1 202:1    0  30G  0 part /
xvdb     202:16   0  30G  0 disk /mnt
xvdf     202:80   0  35G  0 disk
  xvdf1 202:81   0   8G  0 part
```

The root volume, `/dev/xvda1`, is a partition on `/dev/xvda`. Notice that they are both 30 GiB in size. In this case, the partition occupies all of the room on the device, so it does not need resizing.

The volume `/dev/xvdb` is not partitioned at all, so it does not need resizing.

However, `/dev/xvdf1` is an 8 GiB partition on a 35 GiB device and there are no other partitions on the volume. In this case, the partition must be resized in order to use the remaining space on the volume. For more information, see Expanding a Linux Partition (p. 549). After you resize the partition, you can follow the next procedure to extend the file system to occupy all of the space on the partition.

To extend a Linux file system

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 Instance Using SSH (p. 299).
2. Use the **df -h** command to report the existing file system disk space usage. In this example, the `/dev/xvda1` device has already been expanded to 70 GiB, but the `ext4` file system only sees the original 8 GiB size, and the `/dev/xvdf` device has been expanded to 100 GiB, but the XFS file system only sees the original 1 GiB size.

```
[ec2-user ~]$ 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
/dev/xvdf          1014M   33M  982M   4% /mnt
```

3. Use the file system-specific command to resize the file system to the new size of the volume. For a Linux `ext2`, `ext3`, or `ext4` file system, use the following command, substituting the device name that you want to extend.

```
[ec2-user ~]$ sudo 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.
```

For an XFS file system, use the following command, substituting the mount point of the file system (XFS file systems must be mounted to resize them).

```
[ec2-user ~]$ sudo xfs_growfs -d /mnt
meta-data=/dev/xvdf              isize=256    agcount=4, agsize=65536 blks
         =                       sectsz=512   attr=2
data     =                       bsize=4096   blocks=262144, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2             bsize=4096   ascii-ci=0
log      =internal              bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
data blocks changed from 262144 to 26214400
```

> **Note**
> If you receive an `xfsctl failed: Cannot allocate memory` error, you may need to update the Linux kernel on your instance. For more information, refer to your specific operating system documentation.

4. Use the **df -h** command to report the existing file system disk space usage, which should now show the full 70 GiB on the `ext4` file system and 100 GiB on the XFS file system.

```
# df -h
Filesystem             Size  Used Avail Use% Mounted on
/dev/xvda1              69G  951M   68G   2% /
tmpfs                  1.9G     0  1.9G   0% /dev/shm
/dev/xvdf              100G   45M  100G   1% /mnt
```

Deleting the Old Volume

After the new volume has been attached and extended in the instance, you can delete the old volume if it is no longer needed.

To delete the old volume

1. In the Amazon EC2 console, click **Volumes** in the navigation pane.
2. Right-click the old volume and select **Delete Volume**.
3. In the **Delete Volume** dialog box, click **Yes, Delete**.

Expanding a Linux Partition

Some Amazon EC2 root volumes and volumes that are restored from snapshots contain a partition that actually holds the file system and the data. If you think of a volume as a container, a partition is another container inside the volume, and the data resides on the partition. Growing the volume size does not grow the partition; to take advantage of a larger volume, the partition must be expanded to the new size.

Note

Not all volumes restored from snapshots are partitioned, and this procedure may not apply to your volume. You may just need to resize the file system on your volume to make all of the space available. If you are not sure if your volume has a partition that needs resizing, see To check if your volume partition needs resizing (p. 548) for more information.

If the partition you want to expand is not the root partition, then you can simply unmount it and resize the partition from the instance itself. If the partition you need to resize is the root partition for an instance, the process becomes more complicated because you cannot unmount the root partition of a running instance. You have to perform the following procedures on another instance, which is referred to as a *secondary instance*.

Important

The following procedures were written for and tested on Amazon Linux. Other distributions with different tool sets and tool versions may behave differently.

Topics

- Preparing a Linux Root Partition for Expansion (p. 550)
- Expanding a Linux Partition Using parted (p. 551)
- Expanding a Linux Partition Using gdisk (p. 555)
- Returning an Expanded Partition to its Original Instance (p. 558)

Preparing a Linux Root Partition for Expansion

There are several steps that you need to take to expand the root partition of an instance. If the partition you need to expand is not the root partition, then this procedure is not necessary.

To prepare a Linux root partition for expansion

1. If your primary instance is running, stop it. You cannot perform the rest of this procedure on a running instance. For more information, see Stop and Start Your Instance (p. 309).
2. Take a snapshot of your volume. It can be easy to corrupt or lose your data in the following procedures. If you have a fresh snapshot, you can always start over in case of a mistake and your data will still be safe. For more information, see Creating an Amazon EBS Snapshot (p. 560).
3. Record the device name that the volume is attached to. You can find this information on the **Root device** field of the instance's details pane. The value is likely /dev/sda1 or /dev/xvda.
4. Detach the volume from the primary instance. For more information, see Detaching an Amazon EBS Volume from an Instance (p. 544).
5. Attach the volume to another (secondary) instance in the same Availability Zone. For more information, see Attaching an Amazon EBS Volume to an Instance (p. 529). If your Amazon EBS volume is encrypted, you must use a secondary instance that supports Amazon EBS encryption; otherwise, you can use a t2.micro instance for this procedure. For more information, see Supported Instance Types (p. 565). If you do not already have a secondary instance, you will need to launch one. For more information, see Launching an Instance (p. 290).

   Important

   The secondary instance must be running when you attach the volume, and you should not reboot the secondary instance while multiple root volumes are attached; booting an instance with multiple root volumes attached could cause the instance to boot to the wrong volume.

6. Log in to the secondary instance with SSH. For more information, see Connect to Your Linux Instance (p. 299). Continue with the next procedure.

Expanding a Linux Partition Using parted

The **parted** utility is a partition editing tool that is available on most Linux distributions. It can create and edit both MBR partition tables and GPT partition tables. Some versions of **parted** (newer than version 2.1) have limited support for GPT partition tables and they may cause boot issues if their version of **parted** is used to modify boot volumes. You can check your version of **parted** with the `parted --version` command.

If you are expanding a partition that resides on a GPT partitioned device, you should choose to use the **gdisk** utility instead. For more information, see To expand a Linux partition using **gdisk** (p. 555).

To expand a Linux partition using parted

If the partition you need to expand is the root partition, be sure to follow the steps in To prepare a Linux root partition for expansion (p. 550) first.

1. Identify the device that contains the partition that you want to expand. Use the **lsblk** command to list all devices and partitions attached to the instance.

    ```
    [ec2-user ~]$ lsblk
    NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
    xvdf    202:80   0  100G  0 disk
      xvdf1 202:81   0    8G  0 part /mnt
    xvda1   202:1    0   30G  0 disk /
    ```

    In this example, the `xvdf` device has 100 GiB of available storage and it contains an 8 GiB partition.

2. Unmount the partition if it is mounted. Run the **umount** command with the value of `MOUNTPOINT` from the **lsblk** command. In this example, the `MOUNTPOINT` value for the partition is `/mnt`.

    ```
    [ec2-user ~]$ sudo umount /mnt
    ```

3. Take a snapshot of your volume (unless you just took one in the previous procedure). It can be easy to corrupt or lose your data in the following procedures. If you have a fresh snapshot, you can always start over in case of a mistake and your data will still be safe. For more information, see Creating an Amazon EBS Snapshot (p. 560).

4. Run the **parted** command on the device (and not the partition on the device). Remember to add the `/dev/` prefix to the name that **lsblk** outputs.

    ```
    [ec2-user ~]$ sudo parted /dev/xvdf
    GNU Parted 2.1
    Using /dev/xvdf
    Welcome to GNU Parted! Type 'help' to view a list of commands.
    ```

5. Change the **parted** units of measure to sectors.

    ```
    (parted) unit s
    ```

6. Run the **print** command to list the partitions on the device. For certain partition table types, you might be prompted to repair the partition table for the larger volume size. Answer 'Ignore' to any questions about fixing the existing partition table; you will create a new table later.

    ```
    (parted) print
    ```

a.   If you receive the following message, enter 'Ignore' to prevent the backup GPT location from
     changing.

```
Error: The backup GPT table is not at the end of the disk, as it should
 be.  This might mean that another operating
system believes the disk is smaller.  Fix, by moving the backup to the
end (and removing the old backup)?
Fix/Ignore/Cancel? Ignore
```

b.   If you receive the following message, enter 'Ignore' again to keep the space on the drive the
     same.

```
Warning: Not all of the space available to /dev/xvdf appears to be used,
 you can fix the GPT to use all of the
space (an extra 46137344 blocks) or continue with the current setting?
Fix/Ignore? Ignore
```

7.   Examine the output for the total size of the disk, the partition table type, the number of the partition,
     the start point of the partition, and any flags, such as `boot`. For `gpt` partition tables, note the name
     of the partition; for `msdos` partition tables, note the `Type` field (`primary` or `extended`). These values
     are used in the upcoming steps.

     The following is a `gpt` partition table example.

```
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdf: 209715200s
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start  End        Size         File system  Name                   Flags
128     2048s  4095s      2048s                      BIOS Boot Partition
bios_grub
 1      4096s  16777182s  16773087s    ext4         Linux
```

     The following is an `msdos` partition table example.

```
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdg: 104857600s
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start  End        Size         Type     File system  Flags
 1      2048s  35649535s  35647488s    primary  ext3
```

8.   Delete the partition entry for the partition using the number (1) from the previous step.

```
(parted) rm 1
```

9.   Create a new partition that extends to the end of the volume.

(For the `gpt` partition table example) Note the start point and name of partition 1 above. For the `gpt` example, there is a start point of 4096s, and the name `Linux`. Run the **mkpart** command with the start point of partition 1, the name, and 100% to use all of the available space.

```
(parted) mkpart Linux 4096s 100%
```

(For the `msdos` partition table example) Note the start point and the partition type of partition 1 above. For the `msdos` example, there is a start point of 2048s and a partition type of `primary`. Run the **mkpart** command with a primary partition type, the start point of partition 1, and 100% to use all of the available space.

```
(parted) mkpart primary 2048s 100%
```

10. Run the **print** command again to verify your partition.

(For the `gpt` partition table example)

```
(parted) print
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdf: 209715200s
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start  End          Size         File system  Name
Flags
128     2048s  4095s        2048s                      BIOS Boot Partition
bios_grub
 1      4096s  209713151s   209709056s   ext4          Linux
```

(For the `msdos` partition table example)

```
(parted) print
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdg: 104857600s
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start  End          Size         Type     File system  Flags
 1      2048s  104857599s   104855552s   primary  ext3
```

11. Check to see that any flags that were present earlier are still present for the partition that you expanded. In some cases the `boot` flag may be lost. If a flag was dropped from the partition when it was expanded, add the flag with the following command, substituting your partition number and the flag name. For example, the following command adds the `boot` flag to partition 1.

```
(parted) set 1 boot on
```

You can run the **print** command again to verify your change.

12. Run the **quit** command to exit **parted**.

```
(parted) quit
```

> **Note**
> Because you removed a partition and added a partition, **parted** may warn that you may
> need to update `/etc/fstab`. This is only required if the partition number changes.

13. Check the file system to make sure there are no errors (this is required before you may extend the
    file system). Note the file system type from the previous **print** commands. Choose one of the com-
    mands below based on your file system type; if you are using a different file system, consult the
    documentation for that file system to determine the correct check command.

    (For `ext3` or `ext4` file systems)

    ```
    [ec2-user ~]$ sudo e2fsck -f /dev/xvdf1
    e2fsck 1.42.3 (14-May-2012)
    Pass 1: Checking inodes, blocks, and sizes
    Pass 2: Checking directory structure
    Pass 3: Checking directory connectivity
    Pass 4: Checking reference counts
    Pass 5: Checking group summary information
    /: 31568/524288 files (0.4% non-contiguous), 266685/2096635 blocks
    ```

    (For `xfs` file systems)

    ```
    [ec2-user ~]$ sudo xfs_repair /dev/xvdf1
    Phase 1 - find and verify superblock...
    Phase 2 - using internal log
            - zero log...
            - scan filesystem freespace and inode maps...
            - found root inode chunk
    Phase 3 - for each AG...
            - scan and clear agi unlinked lists...
            - process known inodes and perform inode discovery...
            - agno = 0
            - agno = 1
            - agno = 2
            - agno = 3
            - process newly discovered inodes...
    Phase 4 - check for duplicate blocks...
            - setting up duplicate extent list...
            - check for inodes claiming duplicate blocks...
            - agno = 0
            - agno = 1
            - agno = 2
            - agno = 3
    Phase 5 - rebuild AG headers and trees...
            - reset superblock...
    Phase 6 - check inode connectivity...
            - resetting contents of realtime bitmap and summary inodes
            - traversing filesystem ...
            - traversal finished ...
            - moving disconnected inodes to lost+found ...
    Phase 7 - verify and correct link counts...
    done
    ```

14. The next steps differ depending on whether the expanded partition belongs on the current instance
    or if it is the root partition for another instance.

- If this partition belongs on the current instance, remount the partition at the `MOUNTPOINT` identified in Step 2 (p. 551).

```
[ec2-user ~]$ sudo mount /dev/xvdf1 /mnt
```

After you have mounted the partition, extend the file system to use the newly available space by following the procedures in Extending a Linux File System (p. 547).

- If this volume is the root partition for another instance, proceed to the next procedure.

Expanding a Linux Partition Using gdisk

The **gdisk** utility (sometimes called GPT fdisk) is a text-based, menu-driven tool for creating and editing partition tables, and it has better support for GPT partition tables than **parted** in some distributions. Many common Linux distributions (such as Amazon Linux and Ubuntu) provide **gdisk** by default. If your distribution does not provide the **gdisk** command, you can find out how to get it by visiting Obtaining GPT fdisk; in many cases, it is much easier to simply launch an Amazon Linux instance to use as a secondary instance because the **gdisk** command is already available.

To expand a Linux partition using gdisk

If the partition you need to expand is the root partition, be sure to follow the steps in To prepare a Linux root partition for expansion (p. 550) first.

1. Identify the device that contains the partition that you want to expand. Use the **lsblk** command to list all devices and partitions attached to the instance.

```
[ec2-user ~]$ lsblk
NAME     MAJ:MIN RM  SIZE RO MOUNTPOINT
xvdf     202:80   0  100G  0
  xvdf1 202:81   0  9.9G  0 /mnt
xvda1    202:1    0   30G  0 /
```

In this example, the `xvdf` device has 100 GiB of available storage and it contains an 9.9 GiB partition.

2. Unmount the partition if it is mounted. Run the **umount** command with the value of `MOUNTPOINT` from the **lsblk** command. In this example, the `MOUNTPOINT` value for the partition is `/mnt`.

```
[ec2-user ~]$ sudo umount /mnt
```

3. Take a snapshot of your volume (unless you just took one in the previous procedure). It can be easy to corrupt or lose your data in the following procedures. If you have a fresh snapshot, you can always start over in case of a mistake and your data will still be safe. For more information, see Creating an Amazon EBS Snapshot (p. 560).

4. Run the **gdisk** command on the device (and not the partition on the device). Remember to add the `/dev/` prefix to the name that **lsblk** outputs.

```
[ec2-user ~]$ sudo gdisk /dev/xvdf
gdisk /dev/xvdf
GPT fdisk (gdisk) version 0.8.10

Partition table scan:
  MBR: protective
```

```
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.
```

5. Run the **p** command to print the partition table for the device.

6. Examine the output for the disk identifier, partition number, the starting sector, the code for the partition, and the name of the partition.

```
Command (? for help): p
Disk /dev/xvdf: 209715200 sectors, 100.0 GiB
Logical sector size: 512 bytes
Disk identifier (GUID): 947F4655-F3BF-4A1F-8203-A7B30C2A4425
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 20705246
Partitions will be aligned on 2048-sector boundaries
Total free space is 2108 sectors (1.0 MiB)

Number  Start (sector)    End (sector)  Size        Code  Name
   1            2048        20705152  9.9 GiB       EF00  lxroot
```

In the above example the disk identifier is `947F4655-F3BF-4A1F-8203-A7B30C2A4425`, the partition number is `1`, the starting sector is `2048`, the code is `EF00`, and the name is `lxroot`.

7. Because the existing partition table was originally created for a smaller volume, you need to create a new partition table for the larger volume. Run the **o** command to create a new, empty partition table.

```
Command (? for help): o
This option deletes all partitions and creates a new protective MBR.
Proceed? (Y/N): Y
```

8. Use the **n** command to create a new partition on the device. At each prompt, enter the values that you recorded earlier. For the last sector value, use the default value to expand to the entire volume size.

```
Command (? for help): n
Partition number (1-128, default 1): 1
First sector (34-209715166, default = 2048) or {+-}size{KMGTP}: 2048
Last sector (2048-209715166, default = 209715166) or {+-}size{KMGTP}:
209715166
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): EF00
Changed type of partition to 'EFI System'
```

9. Use the **c** command to change the name of the partition to the name of the previous partition.

```
Command (? for help): c
Using 1
Enter name: lxroot
```

10. Use the **x** command to enter the expert command menu.

11. Use the **g** command to change the disk identifier to the original value.

```
Expert command (? for help): g
Enter the disk's unique GUID ('R' to randomize): 947F4655-F3BF-4A1F-8203-
A7B30C2A4425
The new disk GUID is 947F4655-F3BF-4A1F-8203-A7B30C2A4425
```

12. Use the **w** command to write the changes to the device and exit.

```
Expert command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): Y
OK; writing new GUID partition table (GPT) to /dev/xvdf.
The operation has completed successfully.
```

13. Check the file system to make sure there are no errors (this is required before you may extend the file system).

   a. Find the file system type with the following command, substituting the partition you just expanded.

```
[ec2-user ~]$ sudo file -sL /dev/xvdf1
```

   b. Choose one of the commands below based on your file system type; if you are using a different file system, consult the documentation for that file system to determine the correct check command.

   (For `ext3` or `ext4` file systems)

```
[ec2-user ~]$ sudo e2fsck -f /dev/xvdf1
e2fsck 1.42.3 (14-May-2012)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/: 31568/524288 files (0.4% non-contiguous), 266685/2096635 blocks
```

   (For `xfs` file systems)

```
[ec2-user ~]$ sudo xfs_repair /dev/xvdf1
Phase 1 - find and verify superblock...
Phase 2 - using internal log
        - zero log...
        - scan filesystem freespace and inode maps...
        - found root inode chunk
Phase 3 - for each AG...
        - scan and clear agi unlinked lists...
        - process known inodes and perform inode discovery...
        - agno = 0
        - agno = 1
        - agno = 2
```

```
            - agno = 3
            - process newly discovered inodes...
Phase 4 - check for duplicate blocks...
            - setting up duplicate extent list...
            - check for inodes claiming duplicate blocks...
            - agno = 0
            - agno = 1
            - agno = 2
            - agno = 3
Phase 5 - rebuild AG headers and trees...
            - reset superblock...
Phase 6 - check inode connectivity...
            - resetting contents of realtime bitmap and summary inodes
            - traversing filesystem ...
            - traversal finished ...
            - moving disconnected inodes to lost+found ...
Phase 7 - verify and correct link counts...
done
```

14. The next steps differ depending on whether the expanded partition belongs on the current instance or if it is the root partition for another instance.

   - If this partition belongs on the current instance, remount the partition at the MOUNTPOINT identified in Step 2 (p. 555).

   ```
   [ec2-user ~]$ sudo mount /dev/xvdf1 /mnt
   ```

   After you have mounted the partition, extend the file system to use the newly available space by following the procedures in Extending a Linux File System (p. 547).
   - If this volume is the root partition for another instance, proceed to the next procedure.

Returning an Expanded Partition to its Original Instance

If you expanded a root partition from another instance, follow this procedure to return the volume to its original instance.

To return an expanded root partition to its original instance

1. Detach the expanded partition from its secondary instance. For more information, see Detaching an Amazon EBS Volume from an Instance (p. 544).
2. Reattach the volume to the primary instance using the device name that you identified in Step 3 (p. 550) of the preparation procedure (p. 550). For more information, see Attaching an Amazon EBS Volume to an Instance (p. 529).
3. Start the primary instance. For more information, see Stop and Start Your Instance (p. 309).
4. (Optional) If you launched a secondary instance for the sole purpose of expanding the partition, you can terminate the instance to stop incurring charges. For more information, see Terminate Your Instance (p. 315).
5. Connect to your primary instance and extend the file system to use the newly available space by following the procedures in Extending a Linux File System (p. 547).

After you are finished with this expanding the file system, you can create an AMI from the instance that you can use to launch new instances with the desired partition size. For more information, see Amazon Machine Images (AMI) (p. 53).

Amazon EBS Snapshots

An Amazon EBS snapshot is a point-in-time backup copy of an Amazon EBS volume that is stored in Amazon S3. Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. When you delete a snapshot, only the data exclusive to that snapshot is removed. Active snapshots contain all of the information needed to restore your data (from the time the snapshot was taken) to a new Amazon EBS volume.

If you are dealing with snapshots of sensitive data, you should consider encrypting your data manually before taking the snapshot or storing the data on a volume that is enabled with Amazon EBS encryption. For more information, see Amazon EBS Encryption (p. 564).

Contents

- Creating an Amazon EBS Snapshot (p. 560)
- Deleting an Amazon EBS Snapshot (p. 561)
- Copying an Amazon EBS Snapshot (p. 561)
- Viewing Snapshot Information (p. 563)
- Sharing Snapshots (p. 563)

When you create a new Amazon EBS volume, you can create it based on an existing snapshot; the new volume begins as an exact replica of the original volume that was used to create the snapshot. New volumes created from existing Amazon S3 snapshots load lazily in the background, so you can begin using them right away. If your instance accesses a piece of data that hasn't yet been loaded, the volume immediately downloads the requested data from Amazon S3, and then continues loading the rest of the volume's data in the background. For more information about creating snapshots, see Creating an Amazon EBS Snapshot (p. 560).

Snapshots that are taken from encrypted volumes are automatically encrypted. Volumes that are created from encrypted snapshots are also automatically encrypted. Your encrypted volumes and any associated snapshots always remain protected. For more information, see Amazon EBS Encryption (p. 564).

You can share your unencrypted snapshots with specific individuals, or make them public to share them with the entire AWS community. Users with access to your snapshots can create their own Amazon EBS volumes from your snapshot, but your snapshots remain completely intact. For more information about how to share snapshots, see Sharing Snapshots (p. 563). Encrypted snapshots cannot be shared with anyone, because your volume encryption keys and master key are specific to your account. If you need to share your encrypted snapshot data, you can migrate the data to an unencrypted volume and share a snapshot of that volume. For more information, see Migrating Data (p. 566).

Amazon EBS snapshots are constrained to the region in which they are created. Once you have created a snapshot of an Amazon EBS volume, you can use it to create new volumes in the same region. For more information, see Restoring an Amazon EBS Volume from a Snapshot (p. 527). You can also copy snapshots across AWS regions, making it easier to leverage multiple AWS regions for geographical expansion, data center migration and disaster recovery. You can copy any accessible snapshots that are in the `available` state. For more information, see Copying an Amazon EBS Snapshot (p. 561).

Note
Copying snapshots that were encrypted with non-default AWS Key Management Service master keys is not supported at this time.

Creating an Amazon EBS Snapshot

After writing data to an Amazon EBS volume, you can periodically create a snapshot of the volume to use as a baseline for new volumes or for data backup. If you make periodic snapshots of a volume, the snapshots are incremental so that only the blocks on the device that have changed after your last snapshot are saved in the new snapshot. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.

Snapshots occur asynchronously and the status of the snapshot is `pending` until the snapshot is complete.

Snapshots that are taken from encrypted volumes are automatically encrypted. Volumes that are created from encrypted snapshots are also automatically encrypted. Your encrypted volumes and any associated snapshots always remain protected. For more information, see Amazon EBS Encryption (p. 564).

By default, only you can launch volumes from snapshots that you own. However, you can choose to share your unencrypted snapshots with specific AWS accounts or make them public. For more information, see Sharing Snapshots (p. 563). Encrypted snapshots cannot be shared with anyone, because your volume encryption keys and master key are specific to your account. If you need to share your encrypted snapshot data, you can migrate the data to an unencrypted volume and share a snapshot of that volume. For more information, see Migrating Data (p. 566).

When a snapshot is created from a volume with an AWS Marketplace product code, the product code is propagated to the snapshot.

You can take a snapshot of an attached volume that is in use. However, snapshots only capture data that has been written to your Amazon EBS volume at the time the snapshot command is issued. This might exclude any data that has been cached by any applications or the operating system. If you can pause any file writes to the volume long enough to take a snapshot, your snapshot should be complete. However, if you can't pause all file writes to the volume, you should unmount the volume from within the instance, issue the snapshot command, and then remount the volume to ensure a consistent and complete snapshot. You can remount and use your volume while the snapshot status is `pending`.

To create a snapshot for Amazon EBS volumes that serve as root devices, you should stop the instance before taking the snapshot.

To unmount the volume in Linux, use the following command:

```
umount -d device_name
```

Where *device_name* is the device name (for example, `/dev/sdh`).

To create a snapshot using the console

1. Open the Amazon EC2 console.
2. Click **Snapshots** in the navigation pane.
3. Click **Create Snapshot**.
4. In the **Create Snapshot** dialog box, select the volume to create a snapshot for, and then click **Create**.

To create a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- create-snapshot (AWS CLI)
- ec2-create-snapshot (Amazon EC2 CLI)

- New-EC2Snapshot (AWS Tools for Windows PowerShell)

Deleting an Amazon EBS Snapshot

This section describes how to delete a snapshot. Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. When you delete a snapshot, only the data exclusive to that snapshot is removed. Active snapshots contain all of the information needed to restore your data (from the time the snapshot was taken) to a new Amazon EBS volume.

Deleting previous snapshots of a volume do not affect your ability to restore volumes from later snapshots of that volume.

Note

- If you make periodic snapshots of a volume, the snapshots are incremental so that only the blocks on the device that have changed since your last snapshot are saved in the new snapshot. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.
- You cannot delete a snapshot of the root device of an Amazon EBS volume used by a registered AMI. You must first deregister the AMI before you can delete the snapshot. For more information, see Deregistering Your AMI (p. 90).

To delete a snapshot using the console

1. Open the Amazon EC2 console.
2. Click **Snapshots** in the navigation pane.
3. Select a snapshot and then select **Delete** from the **Actions** list.
4. Click **Yes, Delete**.

To delete a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- delete-snapshot (AWS CLI)
- ec2-delete-snapshot (Amazon EC2 CLI)
- Remove-EC2Snapshot (AWS Tools for Windows PowerShell)

Copying an Amazon EBS Snapshot

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with Amazon Elastic Compute Cloud (Amazon EC2) instances. With Amazon EBS, you can create point-in-time snapshots of volumes and store them on Amazon Simple Storage Service (Amazon S3). After you've stored a snapshot in Amazon S3, you can copy it from one AWS region to another, or within the same region, using the Amazon EC2 console, Amazon EC2 CLI, or the API. You can use snapshot copies to create backups of data, to create new Amazon EBS volumes, or to create Amazon Machine Images (AMIs).

Note
Copying snapshots that were encrypted with non-default AWS Key Management Service master keys is not supported at this time.

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 (N. Virginia) region to the US West (Oregon) region, the first snapshot copy of the volume is a full copy. However, subsequent snapshot copies of the same volume transferred between the same regions are incremental. A snapshot copy can only be done incrementally as long as there is one full copy of the same volume available in the destination region.

> **Note**
> To copy an Amazon Relational Database Service (Amazon RDS) snapshot, see Copying a DB Snapshot in the Amazon Relational Database Service User Guide.

When you copy a snapshot, you are only charged for the data transfer and storage used to copy the snapshot data across regions and to store the copied snapshot in the destination region. You are not charged if the snapshot copy fails. However, if you cancel a snapshot copy that is not yet complete, or delete the source snapshot while the copy is in progress, you are charged for the bandwidth of the data transferred. The snapshot is copied across regions using the secure Amazon S3 Copy and the snapshot copy receives a snapshot ID that's different from the original snapshot's ID.

You can use a copy of an Amazon EBS snapshot in the following ways:

- Geographic Expansion: You can launch your applications in a new region.
- Migration: You can migrate an application to a new region, to enable better availability and minimize cost.
- Disaster Recovery: You can back up your data and logs across different geographical locations at regular intervals. In case of disaster, you can restore your applications using point-in-time backups stored in the secondary region. This minimizes data loss and recovery time.

The Amazon EC2 console, Amazon EC2 CLI, and the API are designed to provide an intuitive customer experience. We use the push model in the console design to minimize user clicks for the Amazon EBS snapshot use cases discussed earlier. You can easily initiate a copy from the console by starting with the source region. We use a pull model in the Amazon EC2 CLI and the API, because these experiences factor in how customers use automation. You only need to know the source snapshot ID and source region to initiate the copy using the Amazon EC2 CLI or API.

To copy a snapshot using the Amazon EC2 console

You can create a copy of an Amazon EBS snapshot using the Amazon EC2 console.

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Snapshots**.
3. Select the snapshot to copy, and then select **Copy** from the **Actions** list.
4. In the **Copy Snapshot** dialog box, update the following as necessary:

    - **Destination region**: Select the region where you want to write the copy of the snapshot.
    - **Description**: By default, the description includes information about the source snapshot so that you can identify a copy from the original. You can change this description as necessary.

        > **Note**
        > Copying snapshots that were encrypted with non-default AWS Key Management Service master keys is not supported at this time.

5. Click **Yes, Copy**.

6.   In the **Copy Snapshot** confirmation dialog box, you can click **Snapshots** to go to the **Snapshots** page in the region specified, or click **Close**.

To view the progress of the copy process later, switch to the destination region, and then refresh the **Snapshots** page. Copies in progress are listed at the top of the page.

To copy a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- copy-snapshot (AWS CLI)
- ec2-copy-snapshot (Amazon EC2 CLI)
- Copy-EC2Snapshot (AWS Tools for Windows PowerShell)

Viewing Snapshot Information

This section describes how to view descriptive information about the snapshots that you have created.

To view descriptive snapshot information in the console

1.   Open the Amazon EC2 console.
2.   Click **Snapshots** in the navigation pane.
3.   To reduce the list, select an option from the **Filter** list. For example, to view only your snapshots, select **Owned By Me**.
4.   To view more information about a snapshot, select it.

To view descriptive snapshot information using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-snapshots (AWS CLI)
- ec2-describe-snapshots (Amazon EC2 CLI)
- Get-EC2Snapshot (AWS Tools for Windows PowerShell)

Sharing Snapshots

This section describes how to share your unencrypted Amazon EBS snapshots with your co-workers or others in the AWS community by modifying the permissions of the snapshot. Encrypted snapshots cannot be shared with anyone, because the keys that are used to encrypt your volumes and snapshots are specific to your account. If you need to share your encrypted snapshot data, you can migrate the data to an unencrypted volume and share a snapshot of that volume. For more information, see Migrating Data (p. 566).

Users that you have authorized can quickly use your unencrypted Amazon EBS shared snapshots as the basis for creating their own Amazon EBS volumes. If you choose, you can also make your data available publicly to all AWS users. Users to whom you have granted access can create their own Amazon EBS volumes based on your snapshot and your original snapshot remains intact.

Note
Snapshots are constrained to the region in which they are created. If you would like to share a snapshot with another region, you need to copy the snapshot to that region. For more information about copying snapshots, see Copying an Amazon EBS Snapshot (p. 561).

Important
When you share a snapshot (whether by sharing it with another AWS account or making it public to all), you are giving others access to all the data on your snapshot. Share snapshots only with people with whom you want to share *all* your snapshot data.

To modify snapshot permissions in the console

This procedure will help you to share your unencrypted snapshots. For security reasons, encrypted snapshots cannot be shared or made public.

1. Open the Amazon EC2 console.
2. Click **Snapshots** in the navigation pane.
3. Select a snapshot and then select **Modify Snapshot Permissions** from the **Actions** list.
4. Choose whether to make the snapshot public or to share it with select AWS accounts:

   Important
   Making your snapshot public shares all snapshot data with everyone. Snapshots with AWS Marketplace product codes cannot be made public.

   - To make the snapshot public, select **Public**.
   - To expose the snapshot only to specific AWS accounts, select **Private**, enter the ID of the AWS account (without hyphens) in the **AWS Account Number** field, and click **Add Permission**. Repeat until you've added all the required AWS accounts.

   Click **Save** when you're done.

To view and modify snapshot permissions using the command line

To view the `createVolumePermission` attribute of a snapshot, you can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- describe-snapshot-attribute (AWS CLI)
- ec2-describe-snapshot-attribute (Amazon EC2 CLI)
- Get-EC2SnapshotAttribute (AWS Tools for Windows PowerShell)

To modify the `createVolumePermission` attribute of a snapshot, you can use one of the following commands.

- modify-snapshot-attribute (AWS CLI)
- ec2-modify-snapshot-attribute (Amazon EC2 CLI)
- Edit-EC2SnapshotAttribute (AWS Tools for Windows PowerShell)

Amazon EBS Encryption

Amazon EBS encryption offers you a simple encryption solution for your Amazon EBS volumes without the need for you to build, maintain, and secure your own key management infrastructure. When you create an encrypted EBS volume and attach it to a supported instance type, data stored at rest on the volume, disk I/O, and snapshots created from the volume are all encrypted. The encryption occurs on the servers that host Amazon EC2 instances, providing encryption of data-in-transit from EC2 instances to EBS storage.

Amazon EBS encryption uses AWS Key Management Service (AWS KMS) Customer Master Keys (CMKs) when creating encrypted volumes and any snapshots created from your encrypted volumes. The first time you create an encrypted Amazon EBS volume in a region, a default CMK is created for you automat-

ically. This key is used for Amazon EBS encryption unless you select a CMK that you created separately using AWS Key Management Service. Creating your own CMK gives you more flexibility, including the ability to create, rotate, disable, define access controls, and audit the encryption keys used to protect your data. For more information, see the AWS Key Management Service Developer Guide.

This feature is supported on all Amazon EBS volume types (General Purpose (SSD), Provisioned IOPS (SSD), and Magnetic), and you can expect the same provisioned IOPS performance on encrypted volumes as you would with unencrypted volumes with a minimal effect on latency. You can access encrypted Amazon EBS volumes the same way you access existing volumes; encryption and decryption are handled transparently and they require no additional action from you, your EC2 instance, or your application. Snapshots of encrypted EBS volumes are automatically encrypted, and volumes that are created from encrypted EBS snapshots are also automatically encrypted.

> **Important**
> Encrypted boot volumes are not supported at this time.

The Amazon EBS encryption feature is also extended to snapshots of your encrypted volumes. Snapshots that are taken from encrypted volumes are automatically encrypted. Volumes that are created from encrypted snapshots are also automatically encrypted. Your encrypted volumes and any associated snapshots always remain protected.

Amazon EBS encryption is only available on select instance types. For more information, see Supported Instance Types (p. 565). You can attach both encrypted and unencrypted volumes to a supported instance type. You can use and manage encrypted EBS volumes and snapshots using the AWS Management Console, command line interface (CLI), AWS SDKs, or the Amazon EC2 API.

Topics

- Encryption Key Management (p. 565)
- Supported Instance Types (p. 565)
- Considerations (p. 566)
- Migrating Data (p. 566)

Encryption Key Management

Amazon EBS encryption handles key management for you. Each newly created volume is encrypted with a unique 256-bit key; any snapshots of this volume and any subsequent volumes created from those snapshots also share that key. These keys are protected by our own key management infrastructure, which implements strong logical and physical security controls to prevent unauthorized access. Your data and associated keys are encrypted using the industry-standard AES-256 algorithm.

Amazon's overall key management infrastructure uses Federal Information Processing Standards (FIPS) 140-2 approved cryptographic algorithms and is consistent with National Institute of Standards and Technology (NIST) 800-57 recommendations.

Each AWS account has a regularly rotated unique master key that is stored completely separate from your data, on a system that is surrounded with strong physical and logical security controls. Each encrypted volume (and its subsequent snapshots) is encrypted with a unique volume encryption key that is then encrypted with a region-specific secure master key. The volume encryption keys are used in memory on the server that hosts your EC2 instance; they are never stored on disk in plain text.

Supported Instance Types

Amazon EBS encryption is available on the instance types listed in the table below. These instance types leverage the Intel AES New Instructions (AES-NI) instruction set to provide faster and simpler data protection. You can attach both encrypted and unencrypted volumes to these instance types simultaneously.

Instance Family	Instance Types that Support Amazon EBS encryption
General purpose	`m3.medium` | `m3.large` | `m3.xlarge` | `m3.2xlarge`
Compute optimized	`c3.large` | `c3.xlarge` | `c3.2xlarge` | `c3.4xlarge` | `c3.8xlarge`
Memory optimized	`cr1.8xlarge` | `r3.large` | `r3.xlarge` | `r3.2xlarge` | `r3.4xlarge` | `r3.8xlarge`
Storage optimized	`i2.xlarge` | `i2.2xlarge` | `i2.4xlarge` | `i2.8xlarge`
GPU instances	`g2.2xlarge`

For more information on these instance types, see Instance Type Details.

Considerations

Snapshots that are taken from encrypted volumes are automatically encrypted with the same volume encryption key used to encrypt the volume. Volumes that are created from encrypted snapshots are also automatically encrypted with the same volume encryption key used to create the snapshot. There is no way to directly create an unencrypted volume from an encrypted snapshot or vice versa.

Public or shared snapshots of encrypted volumes are not supported, because other accounts would not be able to decrypt your data.

There is also no way to encrypt an existing volume. However, you can migrate existing data between encrypted volumes and unencrypted volumes. For more information, see To migrate data between encrypted and unencrypted volumes (p. 566).

> **Important**
> Encrypted boot volumes are not supported at this time.

Migrating Data

If you have existing data that you would like to store on an encrypted volume, you need to migrate the data from your unencrypted volume to a new encrypted volume. Likewise, if you have data that currently resides on an encrypted volume that you would like to share with others, you need to migrate the data you want to share from your encrypted volume to a new unencrypted volume.

To migrate data between encrypted and unencrypted volumes

1. Create your destination volume (encrypted or unencrypted, depending on your use case) by following the procedures in Creating an Amazon EBS Volume (p. 526).
2. Attach the destination volume to the instance that hosts the data you would like to migrate. For more information, see Attaching an Amazon EBS Volume to an Instance (p. 529).
3. Make the destination volume available by following the procedures in Making an Amazon EBS Volume Available for Use (p. 531). For Linux instances, you can create a mount point at `/mnt/destination` and mount the destination volume there.
4. Copy the data from your source directory to the destination volume.

   Use the **rsync** command as follows to copy the data from your source to the destination volume. In this example, the source data is located in `/mnt/source` and the destination volume is mounted at `/mnt/destination`.

   ```
   [ec2-user ~]$ sudo rsync -avh -E --progress /mnt/source/ /mnt/destination/
   ```

Amazon EBS Volume Performance

Several factors can affect the performance of Amazon EBS volumes, such as instance configuration, I/O characteristics, workload demand, and storage configuration. After you learn the basics of working with EBS volumes, it's a good idea to look at the I/O performance you require and at your options for increasing EBS performance to meet those requirements.

Topics

Amazon EBS Performance Tips

- When you consider the performance requirements for your EBS storage application, it is important to start with an EC2 configuration that is optimized for EBS and that can handle the bandwidth that your application storage system requires. For more information, see Amazon EC2 Instance Configuration (p. 568).

- When you measure the performance of your EBS volumes, especially with General Purpose (SSD) and Provisioned IOPS (SSD) volumes, it is important to understand the units of measure involved and how performance is calculated. For more information, see I/O Characteristics (p. 569).

- There is a relationship between the maximum performance of your EBS volumes, the amount of I/O you are driving to them, and the amount of time it takes for each transaction to complete. Each of these factors (performance, I/O, and time) affects the others, and different applications are more sensitive to one factor or another. For more information, see Workload Demand (p. 570).

- There is a 5 to 50 percent reduction in IOPS when you first access each block of data on a newly created or restored EBS volume (General Purpose (SSD), Provisioned IOPS (SSD), or Magnetic). You can avoid this performance hit by accessing each block in advance. For more information, see Pre-Warming Amazon EBS Volumes (p. 570).

- General Purpose (SSD) and Provisioned IOPS (SSD) volumes have a throughput limit of 128 MB/s per volume. Some instance types can drive more I/O throughput than you can provision for a single volume. You can join multiple General Purpose (SSD) or Provisioned IOPS (SSD) volumes together in a RAID 0 configuration to use the available bandwidth for these instances. You can also provide redundancy for your volumes with a RAID 1 (mirrored) configuration. For more information, see RAID Configuration (p. 573).

- You can benchmark your storage and compute configuration to make sure you achieve the level of performance you expect to see before taking your application live. For more information, see Benchmark Volumes (p. 576).

- Amazon Web Services provides performance metrics for EBS that you can analyze and view with Amazon CloudWatch and status checks that you can use to monitor the health of your volumes. For more information, see Monitoring the Status of Your Volumes (p. 534).

- Frequent snapshots provide a higher level of data durability, but they may slightly degrade the performance of your application while the snapshot is in progress. This trade off becomes critical when you have data that changes rapidly. Whenever possible, plan for snapshots to occur during off-peak times in order to minimize workload impact. For more information, see Amazon EBS Snapshots (p. 559).

Amazon EC2 Instance Configuration

When you plan and configure EBS volumes for your application, it is important to consider the configuration of the instances that you will attach the volumes to. In order to get the most performance out of your EBS volumes, you should attach them to an instance with enough bandwidth to support your volumes, such as an EBS-optimized instance or an instance with 10 Gigabit network connectivity. This is especially important when you use General Purpose (SSD) or Provisioned IOPS (SSD) volumes, or when you stripe multiple volumes together in a RAID configuration.

Use EBS-Optimized or 10 Gigabit Network Instances

Any performance-sensitive workloads that require minimal variability and dedicated Amazon EC2 to Amazon EBS traffic, such as production databases or business applications, should use General Purpose (SSD) or Provisioned IOPS (SSD) volumes that are attached to an EBS-optimized instance or an instance with 10 Gigabit network connectivity. EC2 instances that do not meet this criteria offer no guarantee of network resources. The only way to ensure sustained reliable network bandwidth between your EC2 instance and your EBS volumes is to launch the EC2 instance as EBS-optimized or choose an instance type with 10 Gigabit network connectivity. To see which instance types include 10 Gigabit network connectivity, see Instance Type Details.

Choose an EC2 Instance with Enough Bandwidth

Launching an instance that is EBS-optimized provides you with a dedicated connection between your EC2 instance and your EBS volume. However, it is still possible to provision EBS volumes that exceed the available bandwidth for certain instance types, especially when multiple volumes are striped in a RAID configuration. The following table shows which instance types are available to be launched as EBS-optimized, the dedicated throughput to Amazon EBS, the maximum amount of IOPS the instance can support if you are using a 16 KB I/O size, and the approximate I/O bandwidth available on that connection in MB/s. Be sure to choose an EBS-optimized instance that provides more dedicated EBS throughput than your application needs; otherwise, the Amazon EBS to Amazon EC2 connection will become a performance bottleneck.

Instance Type	Dedicated EBS Throughput (Mbps)*	Max 16K IOPS**	Max Bandwidth (MB/s)**
c1.xlarge	1,000	8,000	125
c3.xlarge	500	4,000	62.5
c3.2xlarge	1,000	8,000	125
c3.4xlarge	2,000	16,000	250
g2.2xlarge	1,000	8,000	125
i2.xlarge	500	4,000	62.5
i2.2xlarge	1,000	8,000	125
i2.4xlarge	2,000	16,000	250
m1.large	500	4,000	62.5
m1.xlarge	1,000	8,000	125
m2.2xlarge	500	4,000	62.5
m2.4xlarge	1,000	8,000	125

Instance Type	Dedicated EBS Throughput (Mbps)*	Max 16K IOPS**	Max Bandwidth (MB/s)**
`m3.xlarge`	500	4,000	62.5
`m3.2xlarge`	1,000	8,000	125
`r3.xlarge`	500	4,000	62.5
`r3.2xlarge`	1,000	8,000	125
`r3.4xlarge`	2,000	16,000	250

The `m1.large` instance has a maximum 16 KB IOPS value of 4,000, but unless this instance type is launched as EBS-optimized, that value is an absolute best-case scenario and is not guaranteed; to consistently achieve 4,000 16 KB IOPS, you must launch this instance as EBS-optimized. However, if two 4,000 IOPS volumes are attached to an EBS-optimized `m1.large` instance in a RAID 0 configuration, the EC2 to EBS connection bandwidth limit will prevent these volumes from providing the 256 MB/s maximum aggregate throughput available to them. In this case, we must use an EBS-optimized EC2 instance that supports 250 MB/s of throughput, such as the `r3.4xlarge` instance type.

General Purpose (SSD) and Provisioned IOPS (SSD) volumes have a throughput limit of 128 MB/s per volume, which pairs nicely with a 1,000 Mbps EBS-optimized connection. Instance types that offer more than 1,000 Mbps of throughput to Amazon EBS can use more than one General Purpose (SSD) or Provisioned IOPS (SSD) volume to take advantage of the available throughput.

Instance types with 10 Gigabit network connectivity support up to 800 MB/s of throughput and 48,000 16K IOPS for unencrypted Amazon EBS volumes and up to 25,000 16K IOPS for encrypted Amazon EBS volumes. Since the maximum provisioned IOPS value for EBS volumes is 4,000, you can use many EBS volumes simultaneously to reach the level of I/O performance available to these instance types. To see which instance types include 10 Gigabit network connectivity, see Instance Type Details.

You should use EBS-optimized instances when available to get the full performance benefits of Amazon EBS General Purpose (SSD) and Provisioned IOPS (SSD) volumes. For more information, see Amazon EBS–Optimized Instances (p. 125).

I/O Characteristics

On a given volume configuration, certain I/O characteristics drive the performance behavior on the back end. General Purpose (SSD) and Provisioned IOPS (SSD) volumes deliver consistent performance whether an I/O operation is random or sequential, and also whether an I/O operation is to read or write data. I/O size, however, does make an impact on IOPS because of the way they are measured. In order to fully understand how General Purpose (SSD) and Provisioned IOPS (SSD) volumes will perform in your application, it is important to know what IOPS are and how they are measured.

What are IOPS?

IOPS are input/output operations per second. Amazon EBS measures each I/O operation per second (that is 256 KB or smaller) as one IOPS. I/O operations that are larger than 256 KB are counted in 256 KB capacity units. For example, a 1,024 KB I/O operation would count as 4 IOPS. When you provision a 4,000 IOPS volume and attach it to an EBS-optimized instance that can provide the necessary bandwidth, you can transfer up to 4,000 chunks of data per second (provided that the I/O does not exceed the 128 MB/s per volume throughput limit of General Purpose (SSD) and Provisioned IOPS (SSD) volumes).

This configuration could transfer 4,000 32 KB chunks, 2,000 64 KB chunks, or 1,000 128 KB chunks of data per second as well, before hitting the 128 MB/s per volume throughput limit. If your I/O chunks are

very large, you may experience a smaller number of IOPS than you provisioned because you are hitting the volume throughput limit.

For 32 KB or smaller I/O operations, you should see the amount of IOPS that you have provisioned, provided that you are driving enough I/O to keep the drives busy. For smaller I/O operations, you may even see an IOPS value that is higher than what you have provisioned (when measured on the client side), and this is because the client may be coalescing multiple smaller I/O operations into a smaller number of large chunks.

If you are not experiencing the expected IOPS or throughput you have provisioned, ensure that your EC2 bandwidth is not the limiting factor; your instance should be EBS-optimized (or include 10 Gigabit network connectivity) and your instance type EBS dedicated bandwidth should exceed the I/O throughput you intend to drive. For more information, see Amazon EC2 Instance Configuration (p. 568). Another possible cause for not experiencing the expected IOPS is that you are not driving enough I/O to the EBS volumes. For more information, see Workload Demand (p. 570).

Workload Demand

Workload demand plays an important role in getting the most out of your General Purpose (SSD) and Provisioned IOPS (SSD) volumes. In order for your volumes to deliver the amount of IOPS that are available, they need to have enough I/O requests sent to them. There is a relationship between the demand on the volumes, the amount of IOPS that are available to them, and the latency of the request (the amount of time it takes for the I/O operation to complete).

Average Queue Length

The queue length is the number of pending I/O requests for a device. Optimal average queue length will vary for every customer workload, and this value depends on your particular application's sensitivity to IOPS and latency. If your workload is not delivering enough I/O requests to maintain your optimal average queue length, then your volume might not consistently deliver the IOPS that you have provisioned. However, if your workload maintains an average queue length that is higher than your optimal value, then your per-request I/O latency will increase; in this case, you should provision more IOPS for your volume.

To determine the optimal average queue length for your workload, we recommend that you target a queue length between 4 and 8 for volumes with 2,000 to 4,000 provisioned IOPS, or a queue length of 1 for every 500 IOPS provisioned for volumes with fewer than 2,000 provisioned IOPS. Then you can monitor your application performance and tune that value based on your application requirements. For example, a volume with 1,000 provisioned IOPS should target an average queue length of 2 and tune that value up or down to see what performs best for your application.

> **Note**
> Per-request I/O latency may increase with higher average queue lengths.

Latency

Latency is the true end-to-end client time of an I/O operation; in other words, when the client sends a IO, how long does it take to get an acknowledgement from the storage subsystem that the IO read or write is complete. If your I/O latency is higher than you require, check your average queue length to make sure that your application is not trying to drive more IOPS than you have provisioned. You can maintain high IOPS while keeping latency down by maintaining a low average queue length. Consistently driving a greater number of IOPS to a volume than it has available to it (or provisioned) can also cause increased latency times; if your application requires a greater number of IOPS than your volume can provide, you should consider using a larger General Purpose (SSD) volume with a higher base performance level or a Provisioned IOPS (SSD) volume with more provisioned IOPS to achieve faster latencies.

Pre-Warming Amazon EBS Volumes

When you create any new EBS volume (General Purpose (SSD), Provisioned IOPS (SSD), or Magnetic) or restore a volume from a snapshot, the back-end storage blocks are allocated to you immediately.

However, the first time you access a block of storage, it must be either wiped clean (for new volumes) or instantiated from its snapshot (for restored volumes) before you can access the block. This preliminary action takes time and can cause a 5 to 50 percent loss of IOPS for your volume the first time each block is accessed. For most applications, amortizing this cost over the lifetime of the volume is acceptable. Performance is restored after the data is accessed once.

However, you can avoid this performance hit in a production environment by writing to or reading from all of the blocks on your volume before you use it; this process is called *pre-warming*. Writing to all of the blocks on a volume is preferred, but that is not an option for volumes that were restored from a snapshot, because that would overwrite the restored data. For a completely new volume that was created from scratch, you should write to all blocks before using the volume. For a new volume created from a snapshot, you should read all the blocks that have data before using the volume.

Pre-warming Amazon EBS Volumes on Linux

To Pre-Warm a New Volume on Linux

For a new volume, use the **dd** command to write to all blocks on a volume. This procedure will write zeroes to all of the blocks on the device and it will prepare the volume for use. Any existing data on the volume will be lost.

1. Attach the newly created volume to your Linux instance.
2. Use the **lsblk** command to list the block devices on your instance.

    ```
    [ec2-user ~]$ lsblk
    NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
    xvdf   202:80   0   30G  0 disk
    xvda1  202:1    0    8G  0 disk /
    ```

    Here you can see that the new volume, /dev/xvdf, is attached, but not mounted (because there is no path listed under the MOUNTPOINT column).

    > **Important**
    > If your new volume is mounted, unmount it. Step 4 (p. 571) should not be performed on a mounted device.

3. (Optional) If your volume is mounted, unmount it with the following command.

    ```
    [ec2-user ~]$ sudo umount /dev/xvdf
    ```

4. Use the **dd** command to write to all of the blocks on the device. The if (import file) parameter should be set to one of the Linux virtual devices, such as /dev/zero. The of (output file) parameter should be set to the drive you wish to warm. The bs parameter sets the block size of the write operation; for optimal performance, this should be set to 1 MB.

    ```
    [ec2-user ~]$ sudo dd if=/dev/zero of=/dev/xvdf bs=1M
    ```

    > **Note**
    > This step may take several minutes up to several hours, depending on your EC2 instance bandwidth, the IOPS provisioned for the volume, and the size of the volume.

    When the operation is finished, you will see a report of the write operation.

    ```
    dd: writing `/dev/xvdf': No space left on device
    15361+0 records in
    ```

```
15360+0 records out
32212254720 bytes (32 GB) copied, 2449.12 s, 13.2 MB/s
```

Your volume is now ready for use. For more information, see Making an Amazon EBS Volume Available for Use (p. 531).

To Pre-Warm a Volume Restored from a Snapshot on Linux

For volumes that have been restored from snapshots, use the **dd** command to read from (and optionally write back to) all blocks on a volume. All existing data on the volume will be preserved.

1. Attach the newly restored volume to your Linux instance.
2. Use the **lsblk** command to list the block devices on your instance.

```
[ec2-user ~]$ lsblk
NAME  MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvdf  202:80   0  30G  0 disk
xvda1 202:1    0   8G  0 disk /
```

Here you can see that the new volume, `/dev/xvdf`, is attached, but not mounted (because there is no path listed under the `MOUNTPOINT` column).

> **Important**
> If your new volume is mounted, unmount it. Step 4 (p. 572) should not be performed on a mounted device.

3. (Optional) If your volume is mounted, unmount it with the following command.

```
[ec2-user ~]$ sudo umount /dev/xvdf
```

4. Use the **dd** command to read all of the blocks on the device. Choose one of the **dd** commands below, depending on whether you want to pre-warm just the previously written blocks or the entire volume.

*   **Read-only: To pre-warm only previously written blocks** This command pre-warms your existing data and any restored snapshots of volumes that have been previously fully pre-warmed. This command maintains incremental snapshots; however, because this operation is read-only, it does not pre-warm unused space that has never been written to on the original volume.

    The `if` (input file) parameter should be set to the drive you wish to warm. The `of` (output file) parameter should be set to the Linux null virtual device, `/dev/null`. The `bs` parameter sets the block size of the read operation; for optimal performance, this should be set to 1 MB.

```
[ec2-user ~]$ sudo dd if=/dev/xvdf of=/dev/null bs=1M
```

*   **Read-then-write-back: To pre-warm an entire restored volume** This command reads each block from the device and then writes it back at the same location, which pre-warms the entire volume, including unused space that has never been pre-warmed. Because this command writes to every block on the device, the next snapshot on a volume that is pre-warmed this way will require a full snapshot.

    The `if` (input file) parameter should be set to the drive you wish to warm. The `of` (output file) parameter should also be set to the drive you wish to warm. The `conv=notrunc` parameter instructs **dd** to overwrite each output file block instead of deleting it. The `bs` parameter sets the block size of the read operation; for optimal performance, this should be set to 1 MB.

```
[ec2-user ~]$ sudo dd if=/dev/xvdf of=/dev/xvdf conv=notrunc bs=1M
```

Note
This step may take several minutes up to several hours, depending on your EC2 instance
bandwidth, the IOPS provisioned for the volume, and the size of the volume.

When the operation is finished, you will see a report of the read operation.

```
15360+0 records in
15360+0 records out
32212254720 bytes (32 GB) copied, 2480.02 s, 13.0 MB/s
```

Your volume is now ready for use. For more information, see Making an Amazon EBS Volume
Available for Use (p. 531).

RAID Configuration

With Amazon EBS, you can use any of the standard RAID configurations that you can use with a tradi-
tional bare metal server, as long as that particular RAID configuration is supported by the operating system
for your instance. This is because all RAID is accomplished at the software level. For greater I/O perform-
ance than you can achieve with a single volume, RAID 0 can stripe multiple volumes together; for on-in-
stance redundancy, RAID 1 can mirror two volumes together.

Note
Amazon EBS volume data is replicated across multiple servers in an Availability Zone to prevent
the loss of data from the failure of any single component. This replication makes Amazon EBS
volumes ten times more reliable than typical commodity disk drives. For more information, see
Amazon EBS Availability and Durability in the Amazon EBS product detail pages.

This topic provides basic RAID setup examples. For much more detailed information on RAID configuration,
performance, and recovery, see the Linux RAID Wiki at https://raid.wiki.kernel.org/index.php/Linux_Raid.

The following table compares the common RAID 0 and RAID 1 options.

Configuration	Use	Advantages	Disadvantages
RAID 0	When I/O performance is more important than fault tolerance; for example, as in a heavily used database (where data replication is already set up separately).	I/O is distributed across the volumes in a stripe. If you add a volume, you get the straight addition of throughput.	Performance of the stripe is limited to the worst performing volume in the set. Loss of a single volume results in a complete data loss for the array.
RAID 1	When fault tolerance is more important than I/O performance; for example, as in a critical application.	Safer from the standpoint of data durability.	Does not provide a write performance improvement; requires more Amazon EC2 to Amazon EBS bandwidth than non-RAID configurations because the data is written to multiple volumes simultaneously.

Important
RAID 5 and RAID 6 are not recommended for Amazon EBS because the parity write operations of these RAID modes consume some of the IOPS available to your volumes. Depending on the configuration of your RAID array, these RAID modes provide 20-30% fewer usable IOPS than a RAID 0 configuration. Increased cost is a factor with these RAID modes as well; when using identical volume sizes and speeds, a 2-volume RAID 0 array can outperform a 4-volume RAID 6 array that costs twice as much.

Creating a RAID 0 array allows you to achieve a higher level of performance for a file system than you can provision on a single Amazon EBS volume. A RAID 1 array offers a "mirror" of your data for extra redundancy. Before you perform this procedure, you need to decide how large your RAID array should be and how many IOPS you want to provision.

The resulting size of a RAID 0 array is the sum of the sizes of the volumes within it, and the bandwidth is the sum of the available bandwidth of the volumes within it. The resulting size and bandwidth of a RAID 1 array is equal to the size and bandwidth of the volumes in the array. For example, two 500 GiB Amazon EBS volumes with 4,000 provisioned IOPS each will create a 1 TiB RAID 0 array with an available bandwidth of 8,000 IOPS and 256 MB/s of throughput or a 500 GiB RAID 1 array with an available bandwidth of 4,000 IOPS and 128 MB/s of throughput.

Creating a RAID Array on Linux

Use the following procedure to create the RAID array. Note that you can get directions for Windows instances from Creating a RAID Array on Windows in the *Amazon EC2 User Guide for Microsoft Windows Instances*.

To create a RAID array on Linux

1. Create the Amazon EBS volumes for your array. For more information, see Creating an Amazon EBS Volume (p. 526).

    Important
    Create volumes with identical size and IOPS performance values for your array. Make sure you do not create an array that exceeds the available bandwidth of your EC2 instance. For more information, see Amazon EC2 Instance Configuration (p. 568).

2. Attach the Amazon EBS volumes to the instance that you want to host the array. For more information, see Attaching an Amazon EBS Volume to an Instance (p. 529).

3. (Optional) Pre-warm your volumes. Amazon EBS volumes can experience between a 5 to 50 percent performance hit the first time you access a block on the volume. Pre-warming allows you to access each block on the volumes before you need to use them so that you can achieve full performance in a production workload. For more information, see Pre-Warming Amazon EBS Volumes (p. 570).

4. Use the **mdadm** command to create a logical RAID device from the newly attached Amazon EBS volumes. Substitute the number of volumes in your array for *number_of_volumes* and the device names for each volume in the array (such as `/dev/xvdf`) for *device_name*.

    Note
    You can list the devices on your instance with the **lsblk** command to find the device names.

    (RAID 0 only) To create a RAID 0 array, execute the following command (note the `--level=stripe` option to stripe the array):

    ```
    [ec2-user ~]$ sudo mdadm --create --verbose /dev/md0 --level=stripe --raid-
    devices=number_of_volumes device_name1 device_name2
    ```

    (RAID 1 only) To create a RAID 1 array, execute the following command (note the `--level=mirror` option to mirror the array):

```
[ec2-user ~]$ sudo mdadm --create --verbose /dev/md0 --level=mirror --raid-
devices=number_of_volumes device_name1 device_name2
```

5. Create a file system on your RAID device. For example, to create an `ext4` file system, execute the following command:

```
[ec2-user ~]$ sudo mkfs.ext4 /dev/md0
```

Depending on the requirements of your application or the limitations of your operating system, you can use a different file system type, such as ext3 or XFS.

6. Create a mount point for your RAID array.

```
[ec2-user ~]$ sudo mkdir /mnt/md0
```

7. Finally, mount the RAID device on the mount point that you created:

```
[ec2-user ~]$ sudo mount -t ext4 /dev/md0 /mnt/md0
```

Your RAID device is now ready for use.

8. (Optional) To mount this Amazon EBS volume on every system reboot, add an entry for the device to the `/etc/fstab` file.

   a. Create a backup of your `/etc/fstab` file that you can use if you accidentally destroy or delete this file while you are editing it.

   ```
   [ec2-user ~]$ sudo cp /etc/fstab /etc/fstab.orig
   ```

   b. Open the `/etc/fstab` file using your favorite text editor, such as **nano** or **vim**.

   c. Add a new line to the end of the file for your volume using the following format.

   ```
   device_name  mount_point  file_system_type  fs_mntops  fs_freq  fs_passno
   ```

   The last three fields on this line are the file system mount options, the dump frequency of the file system, and the order of file system checks done at boot time. If you don't know what these values should be, then use the values in the example below for them (`defaults,nofail 0 2`). For more information on `/etc/fstab` entries, see the **fstab** manual page (by entering **man fstab** on the command line). For example, to mount the ext4 file system on the device `/dev/xvdf` at the mount point `/data`, add the following entry to `/etc/fstab`.

   > **Note**
   > If you ever intend to boot your instance without this volume attached (for example, so this volume could move back and forth between different instances), you should add the `nofail` mount option that allows the instance to boot even if there are errors in mounting the volume. Debian derivatives, such as Ubuntu, must also add the `noboot-wait` mount option.

   ```
   /dev/md0      /mnt/md0   ext4     defaults,nofail       0        2
   ```

   d. After you've added the new entry to `/etc/fstab`, you need to check that your entry works. Run the **sudo mount -a** command to mount all file systems in `/etc/fstab`.

```
[ec2-user ~]$ sudo mount -a
```

If the previous command does not produce an error, then your `/etc/fstab` file is OK and your file system will mount automatically at the next boot. If the command does produce any errors, examine the errors and try to correct your `/etc/fstab`.

> **Warning**
> Errors in the `/etc/fstab` file can render a system unbootable. Do not shut down a system that has errors in the `/etc/fstab` file.

e. (Optional) If you are unsure how to correct `/etc/fstab` errors, you can always restore your backup `/etc/fstab` file with the following command.

```
[ec2-user ~]$ sudo mv /etc/fstab.orig /etc/fstab
```

Benchmark Volumes

This section demonstrates how you can test the performance of Amazon EBS volumes by simulating workloads similar to those of a database application. The process is as follows:

1. Launch an EBS-optimized instance
2. Create new Amazon EBS volumes
3. Attach the volumes to your EBS-optimized instance
4. Create a RAID array from the volumes, then format and mount it
5. Install a tool to benchmark I/O performance
6. Benchmark the I/O performance of your volumes
7. Delete your volumes and terminate your instance so that you don't continue to incur charges

Set Up Your Instance

To get optimal performance from General Purpose (SSD) and Provisioned IOPS (SSD) volumes, we recommend that you use an EBS-optimized instance. EBS-optimized instances deliver dedicated throughput between Amazon EC2 and Amazon EBS, with options between 500 and 2,000 Mbps, depending on the instance type.

To create an EBS-optimized instance, select **Launch as an EBS-Optimized instance** when launching the instance using the EC2 console, or specify **--ebs-optimized** when using the command line. Be sure that you launch one of the instance types that supports this option. For the example tests in this topic, we recommend that you launch an `m1.xlarge` instance. For more information, see Amazon EBS–Optimized Instances (p. 125).

To create a General Purpose (SSD) volume, select **General Purpose (SSD)** when creating the volume using the EC2 console, or specify **--type gp2** when using the command line. To create a Provisioned IOPS (SSD) volume, select **Provisioned IOPS (SSD)** when creating the volume using the EC2 console, or specify **--type io1 --iops** *iops* when using the command line. For information about attaching these volumes to your instance, see Attaching an Amazon EBS Volume to an Instance (p. 529).

For the example tests, we recommend that you create a RAID array with 6 volumes, which offers a high level of performance. Because you are charged by the gigabytes used (and the number of provisioned IOPS for Provisioned IOPS (SSD) volumes), not the number of volumes, there is no additional cost for creating multiple, smaller volumes and using them to create a stripe set. If you're using Oracle ORION

to benchmark your volumes, it can simulate striping the same way that Oracle ASM does, so we recommend that you let ORION do the striping. If you are using a different benchmarking tool, you'll need to stripe the volumes yourself.

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_vol0 && sudo mkfs.xfs /dev/md0 &&
sudo mount -t xfs /dev/md0 /media/p_iops_vol0 && sudo chown ec2-user:ec2-user
/media/p_iops_vol0/
```

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
```

Important
Unless you pre-warm the volume, you might see between a 5 to 50 percent reduction in IOPS when you first access it.

Install Benchmark Tools

The following are among the possible tools you can install and use to benchmark the performance of Amazon EBS volumes.

Tool	Description
fio	For benchmarking I/O performance. (Note that fio has a dependency on libaio-devel.)
Oracle Orion Calibration Tool	For calibrating the I/O performance of storage systems to be used with Oracle databases.

Example Benchmarking Commands

These benchmarking tools support a wide variety of test parameters. You should use commands that approximate the workloads your volumes will support. These commands are intended as examples to help you get started.

Run the following commands on an EBS-optimized instance with attached Amazon EBS volumes that have been pre-warmed.

When you are finished testing your volumes, see these topics for help cleaning up: Deleting an Amazon EBS Volume (p. 545) and Terminate Your Instance (p. 315).

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_vol0
--name fio_test_file --direct=1 --rw=randwrite --bs=16k --size=1G
--numjobs=16 --time_based --runtime=180 --group_reporting --norandommap
```

The following command performs 16 KB random read operations.

```
$ fio --directory=/media/p_iops_vol0
--name fio_test_file --direct=1 --rw=randread --bs=16k --size=1G
--numjobs=16 --time_based --runtime=180 --group_reporting --norandommap
```

For more information about interpreting the results, see this tutorial: Inspecting disk IO performance with fio.

Oracle ORION Commands

Run ORION on the Amazon EBS volumes, having it simulate Oracle ASM striping instead of providing it with a stripe set that uses Windows striping.

In the directory where you installed ORION, create a file, `piops_test.lun`, to specify the volumes for your stripe set. The following example file specifies six volumes to be striped.

```
\\.\D:
\\.\E:
\\.\F:
\\.\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.

Amazon EBS API and Command Overview

The following table summarizes the available commands for Amazon EBS and corresponding API actions for creating and using Amazon EBS volumes.

Command/Action	Description
attach-volume (AWS CLI) ec2-attach-volume (Amazon EC2 CLI) AttachVolume	Attaches the specified volume to a specified instance, exposing the volume using the specified device name. A volume can be attached to only a single instance at any time. The volume and instance must be in the same Availability Zone. The instance must be in the `running` or `stopped` state.
copy-snapshot (AWS CLI) ec2-copy-snapshot (Amazon EC2 CLI) CopySnapshot	Copies a point-in-time snapshot of an Amazon EBS volume and stores it in Amazon S3. You can copy the snapshot within the same region or from one region to another. You can use the snapshot to create new Amazon EBS volumes or AMIs.
create-snapshot (AWS CLI) ec2-create-snapshot (Amazon EC2 CLI) CreateSnapshot	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.
create-volume (AWS CLI) ec2-create-volume (Amazon EC2 CLI) CreateVolume	Creates a new Amazon EBS volume using the specified size and type, or based on a previously created snapshot.
delete-snapshot (AWS CLI) ec2-delete-snapshot (Amazon EC2 CLI) DeleteSnapshot	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.
delete-volume (AWS CLI) ec2-delete-volume (Amazon EC2 CLI) DeleteVolume	Deletes the specified volume. The command does not delete any snapshots that were created from the volume.
describe-snapshot-attribute (AWS CLI) ec2-describe-snapshot-attribute (Amazon EC2 CLI) DescribeSnapshotAttribute	Describes attributes for a snapshot.
describe-snapshots (AWS CLI) ec2-describe-snapshots (Amazon EC2 CLI) DescribeSnapshots	Describes the specified snapshot. Describes all snapshots, including their source volume, snapshot initiation time, progress (percentage complete), and status (`pending`, `completed`, and so on.).

Command/Action	Description
describe-volume-attribute (AWS CLI) ec2-describe-volume-attribute (Amazon EC2 CLI) DescribeVolumeAttribute	Describes an attribute of a volume.
describe-volume-status (AWS CLI) ec2-describe-volume-status (Amazon EC2 CLI) DescribeVolumeStatus	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.
describe-volumes (AWS CLI) ec2-describe-volumes (Amazon EC2 CLI) DescribeVolumes	Describes your volumes, including size, volume type, source snapshot, Availability Zone, creation time, status (`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.
detach-volume (AWS CLI) ec2-detach-volume (Amazon EC2 CLI) DetachVolume	Detaches the specified volume from the instance it's attached to. This command does not delete the volume. The volume can be attached to another instance and will have the same data as when it was detached.
enable-volume-io (AWS CLI) ec2-enable-volume-io (Amazon EC2 CLI) EnableVolumeIO	Enables I/O operations for a volume that had I/O operations disabled because the data on the volume was potentially inconsistent.
modify-snapshot-attribute (AWS CLI) ec2-modify-snapshot-attribute (Amazon EC2 CLI) ModifySnapshotAttribute	Modifies permissions for a snapshot (i.e., who can create volumes from the snapshot). You can specify one or more AWS accounts, or specify `all` to make the snapshot public.
modify-volume-attribute (AWS CLI) ec2-modify-volume-attribute (Amazon EC2 CLI) ModifyVolumeAttribute	Modifies a volume's attributes to determine whether a volume should be automatically enabled for I/O operations.

Command/Action	Description
reset-snapshot-attribute (AWS CLI) ec2-reset-snapshot-attribute (Amazon EC2 CLI) ResetSnapshotAttribute	Resets permission settings for the specified snapshot.

Amazon EC2 Instance Store

Many Amazon EC2 instance types can access disk storage from disks that are physically attached to the host computer. This disk storage is referred to as *instance store*.

Contents

Instance Storage Concepts

An instance store provides temporary block-level storage for use with an instance. The size of an instance store ranges from 900 MiB to up to 48 TiB, and varies by instance type. Larger instance types have larger instance stores. Some smaller instance families, such as T2 and T1, do not support instance store volumes at all and they use Amazon EBS exclusively for storage. For more information, see Instance Stores Available on Instance Types (p. 582).

An instance store consists of one or more instance store volumes. When you launch an instance store-backed AMI, each instance store volume available to the instance is automatically mapped. When you launch an Amazon EBS-backed AMI, instance store volumes must be configured using block device mapping at launch time (with either the default block device mapping for the chosen AMI or manually using the console or the CLI or SDK tools). Volumes must be formatted and mounted on the running in-stance before they can be used. Some AMIs (such as Ubuntu and Amazon Linux) that use the `cloud-init` utilities during the initial boot cycle may format and mount a single instance store volume, but this varies by AMI. By default, instances launched from an Amazon EBS-backed AMI have no mounted instance store volumes. Instances launched from an instance store-backed AMI have a mounted instance store volume for the virtual machine's root device volume (the size of this volume varies by AMI, but the maximum size is 10 GiB) in addition to the instance store volumes included with the instance type. For more information about instance store-backed AMIs and Amazon EBS-backed AMIs, see Storage for the Root Device (p. 55).

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.

Host Computer 1   Host Computer 2
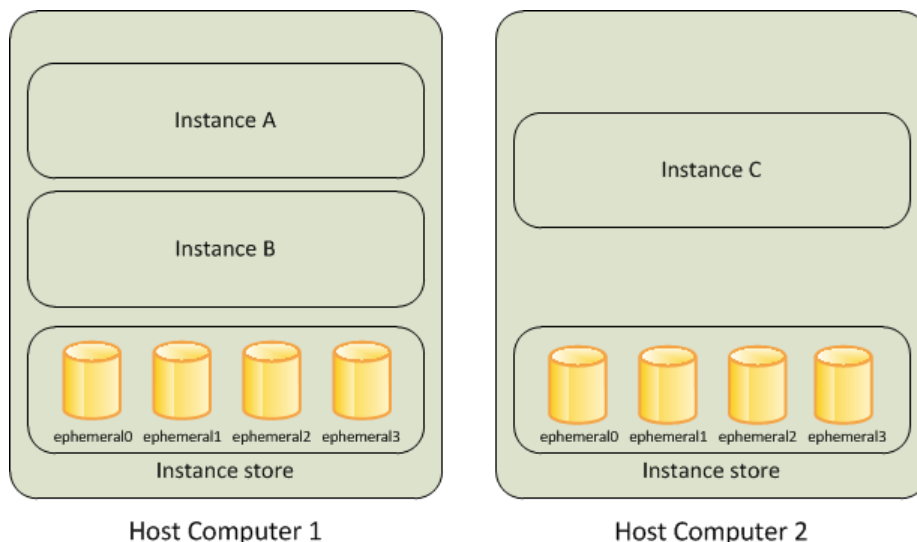
The data in an instance store persists only during the lifetime of its associated instance. If an instance reboots (intentionally or unintentionally), data in the instance store persists. However, data on instance store volumes is lost under the following circumstances:

- 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. 519).

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. 588).

Instance Stores Available on Instance Types

Amazon EC2 instances are divided into different instance types, which determine the size of the instance store available on the instance by default. When you launch an instance, you can specify an instance type or use the default instance type, which is an `m1.small` instance.

The instance type also determines the type of hardware for your instance store volumes. Some instance types use solid state drives (SSD) to deliver very high random I/O performance. This is a good option when you need storage with very low latency, but you don't need it to persist when the instance terminates, or you can take advantage of fault tolerant architectures. For more information see HI1 Instances (p. 110).

The following table shows the instance types along with the size and quantity of the instance store volumes available to each instance type; these instance store volumes are included as part of the instance's hourly cost.

Instance Type	Instance Store Volumes
`c1.medium`	1 x 350 GB†
`c1.xlarge`	4 x 420 GB (1680 GB)

Instance Type	Instance Store Volumes
c3.large	2 x 16 GB SSD (32 GB)
c3.xlarge	2 x 40 GB SSD (80 GB)
c3.2xlarge	2 x 80 GB SSD (160 GB)
c3.4xlarge	2 x 160 GB SSD (320 GB)
c3.8xlarge	2 x 320 GB SSD (640 GB)
cc2.8xlarge	4 x 840 GB (3360 GB)
cg1.4xlarge	2 x 840 GB (1680 GB)
cr1.8xlarge	2 x 120 GB SSD (240 GB)
g2.2xlarge	1 x 60 GB SSD
hi1.4xlarge	2 x 1024 GB SSD (2048 GB)
hs1.8xlarge	24 x 2048 GB (49 TB)
i2.xlarge	1 x 800 GB SSD
i2.2xlarge	2 x 800 GB SSD (1600 GB)
i2.4xlarge	4 x 800 GB SSD (3200 GB)
i2.8xlarge	8 x 800 GB SSD (6400 GB)
m1.small	1 x 160 GB†
m1.medium	1 x 410 GB
m1.large	2 x 420 GB (840 GB)
m1.xlarge	4 x 420 GB (1680 GB)
m2.xlarge	1 x 420 GB
m2.2xlarge	1 x 850 GB
m2.4xlarge	2 x 840 GB (1680 GB)
m3.medium	1 x 4 GB SSD
m3.large	1 x 32 GB SSD
m3.xlarge	2 x 40 GB SSD (80 GB)
m3.2xlarge	2 x 80 GB SSD (160 GB)
r3.large	1 x 32 GB SSD
r3.xlarge	1 x 80 GB SSD
r3.2xlarge	1 x 160 GB SSD
r3.4xlarge	1 x 320 GB SSD
r3.8xlarge	2 X 320 GB SSD (640 GB)
t1.micro	None (use Amazon EBS volumes)

Instance Type	Instance Store Volumes
`t2.micro`	None (use Amazon EBS volumes)
`t2.small`	None (use Amazon EBS volumes)
`t2.medium`	None (use Amazon EBS volumes)

† The `c1.medium` and `m1.small` instance types also include a 900 MB instance store swap volume, which may not be automatically enabled at boot time. For more information, see Instance Store Swap Volumes (p. 584).

Instance Store Swap Volumes

Swap space in Linux can be used when a system requires more memory than it has been physically allocated. When swap space is enabled, Linux systems can swap infrequently used memory pages from physical memory to swap space (either a dedicated partition or a swap file in an existing file system) and free up that space for memory pages that require high speed access.

Note
Using swap space for memory paging is not as fast or efficient as using RAM. If your workload is regularly paging memory into swap space, you should consider migrating to a larger instance type with more RAM. For more information, see Resizing Your Instance (p. 128).

The `c1.medium` and `m1.small` instance types have a limited amount of physical memory to work with, and they are given a 900 MB swap volume at launch time to act as virtual memory for Linux AMIs. Although the Linux kernel sees this swap space as a partition on the root device, it is actually a separate instance store volume, regardless of your root device type.

Amazon Linux AMIs automatically enable and use this swap space, but your AMI may require some additional steps to recognize and use this swap space. To see if your instance is using swap space, you can use the **swapon -s** command.

```
[ec2-user@ip-12-34-56-78 ~]$ swapon -s
Filename                                Type          Size    Used    Priority
/dev/xvda3                              partition     917500  0       -1
```

The above instance has a 900 MB swap volume attached and enabled. If you don't see a swap volume listed with this command, you may need to enable swap space for the device. Check your available disks using the **lsblk** command.

```
[ec2-user@ip-12-34-56-78 ~]$ lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda1 202:1    0    8G  0 disk /
xvda3 202:3    0  896M  0 disk
```

Here, the swap volume `xvda3` is available to the instance, but it is not enabled (notice that the `MOUNTPOINT` field is empty). You can enable the swap volume with the **swapon** command.

Note
You need to prepend `/dev/` to the device name listed by **lsblk**. Your device may be named differently, such as `sda3`, `sde3`, or `xvde3`. Use the device name for your system in the command below.

```
[ec2-user@ip-12-34-56-78 ~]$ sudo swapon /dev/xvda3
```

Now the swap space should show up in **lsblk** and **swapon -s** output.

```
[ec2-user@ip-12-34-56-78 ~]$ lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda1 202:1   0    8G  0 disk /
xvda3 202:3   0  896M  0 disk [SWAP]
[ec2-user@ip-12-34-56-78 ~]$ swapon -s
Filename                                Type          Size    Used    Priority
/dev/xvda3                              partition     917500  0        -1
```

You will also need to edit your /etc/fstab file so that this swap space is automatically enabled at every system boot.

```
[ec2-user@ip-12-34-56-78 ~]$ sudo vim /etc/fstab
```

Append the following line to your /etc/fstab file (using the swap device name for your system):

```
/dev/xvda3        none    swap    sw  0       0
```

To use an instance store volume as swap space

Any instance store volume can be used as swap space. For example, the m3.medium instance type includes a 4 GB SSD instance store volume that is appropriate for swap space. If your instance store volume is much larger (for example, 350 GB), you may consider partitioning the volume with a smaller swap partition of 4-8 GB and the rest for a data volume.

1. List the block devices attached to your instance to get the device name for your instance store volume.

    ```
    [ec2-user ~]$ lsblk -p
    NAME        MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
    /dev/xvdb  202:16   0   4G  0 disk /media/ephemeral0
    /dev/xvda1 202:1    0   8G  0 disk /
    ```

    In this example, the instance store volume is /dev/xdvb. Because this is an Amazon Linux instance, the instance store volume is formatted and mounted at /media/ephemeral0; not all Linux operating systems do this automatically.

2. (Optional) If your instance store volume is mounted (it will list a MOUNTPOINT in the **lsblk** command output), you need to unmount it with the following command.

    ```
    [ec2-user ~]$ sudo umount /dev/xvdb
    ```

3. Set up a Linux swap area on the device with the **mkswap** command.

    ```
    [ec2-user ~]$ sudo mkswap /dev/xvdb
    mkswap: /dev/xvdb: warning: wiping old ext3 signature.
    Setting up swapspace version 1, size = 4188668 KiB
    no label, UUID=b4f63d28-67ed-46f0-b5e5-6928319e620b
    ```

4. Enable the new swap space.

    ```
    [ec2-user ~]$ sudo swapon /dev/xvdb
    ```

5. Verify that the new swap space is being used.

```
[ec2-user ~]$ swapon -s
Filename     Type  Size Used Priority
/dev/xvdb                               partition 4188668 0 -1
```

6. Edit your `/etc/fstab` file so that this swap space is automatically enabled at every system boot.

```
[ec2-user ~]$ sudo vim /etc/fstab
```

If your `/etc/fstab` file has an entry for `/dev/xvdb` (or `/dev/sdb`) change it to match the line below; if it does not have an entry for this device, append the following line to your `/etc/fstab` file (using the swap device name for your system):

```
/dev/xvdb        none    swap    sw  0       0
```

Instance Store Device Names

Within an instance store, instance store volumes are exposed as block devices. The virtual devices for instance store volumes are `ephemeral[0-23]`. Instance types that support one instance store volume have `ephemeral0`. Instance types that support two instance store volumes have `ephemeral0` and `ephemeral1`. Instance types that support four instance store volumes have `ephemeral0`, `ephemeral1`, `ephemeral2`, and `ephemeral3`, and so on.

Many instance store volumes are pre-formatted with the ext3 file system. SSD-based instance store volumes that support TRIM instruction are not pre-formatted with any file system. However, you can format volumes with the file system of your choice after you launch your instance.

Each entry in a block device mapping consists of a device name and the volume that it's mapped to. The instance store volumes are available to the instance, but you can't access them until they are mounted. 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.

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. 582).

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. 591)

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. Launch an instance that supports at least two instance store volumes with this AMI and add block device mapping entries for `ephemeral0` and `ephemeral1`.

   For more information, see To add volumes to an instance (p. 598).
3. Connect to the instance.
4. Verify the instance stores currently mounted on the disk.
5. Notice a 10GiB root partition mounted at `/` and a 420 GiB mounted on an `/media/ephemeral0` 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 `/dev/sdc` (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 an Amazon EBS-Backed Linux AMI (p. 74).

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 `m1.large` 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 `ephemeral0` device. 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 mount the other 420GiB: a. Create a directory on which to mount the device. b. Mount the device on the newly created directory.
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. 591).

Adding Instance Store Volumes to an AMI

Amazon EBS-backed AMIs don't include an instance store by default. However, you might want instances launched from your Amazon EBS-backed AMIs to include instance store volumes.

After you add instance store volumes to an AMI, any instance you launch from the AMI includes these instance store volumes. You can confirm that the instance store devices are available from within the instance itself using instance metadata. For more information, see Viewing the Instance Block Device Mapping for Instance Store Volumes (p. 601).

> **Note**
> For M3 instances, you must specify instance store volumes in the block device mapping for the instance when you launch it. When you launch an M3 instance, instance store volumes specified in the block device mapping for the AMI may be ignored if they are not specified as part of the instance block device mapping.

You can create an AMI that includes instance store volumes using the console or the command line.

To add instance store volumes to an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**.
3. Select an instance, click **Actions**, and then select **Create Image**.
4. In the **Create Image** dialog, add a meaningful name and description to your image.
5. For each instance store volume, Click **Add New Volume**, select an instance store volume from the **Type** list and a device name from **Device**.
6. Click **Create Image**.

To add instance store volumes to an AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- create-image or register-image (AWS CLI)
- ec2-create-image ec2-register (Amazon EC2 CLI)

For more information, see Block Device Mapping (p. 591).

Optimizing Disk Performance

Because of the way that Amazon EC2 virtualizes disks, the first write to any location on an instance store volume performs more slowly than subsequent writes. For most applications, amortizing this cost over the lifetime of the instance is acceptable. However, if you require high disk performance, we recommend that you pre-warm your drives by writing once to every drive location before production use.

> **Note**
> The I2, R3, and HI1 instance types use direct-attached solid state drives that provide maximum performance at launch time, without pre-warming.

If you require greater flexibility in latency or throughput, we recommend using Amazon EBS.

To pre-warm the instance storage volumes, 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. 582).

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.

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. 519).

Objects are the fundamental entities stored in Amazon S3. Every object stored in Amazon S3 is contained in a bucket. Buckets organize the Amazon S3 namespace at the highest level and identify the account responsible for that storage. Amazon S3 buckets are similar to Internet domain names. Objects stored in the buckets have a unique key value and are retrieved using a HTTP URL address. For example, if an object with a key value `/photos/mygarden.jpg` is stored in the `myawsbucket` bucket, then it is addressable using the URL `http://myawsbucket.s3.amazonaws.com/photos/mygarden.jpg`.

For more information about the features of Amazon S3, see the Amazon S3 product page.

Amazon S3 and Amazon EC2

Given the benefits of Amazon S3 for storage, you may decide to use this service to store files and data sets for use with EC2 instances. There are several ways to move data to and from Amazon S3 to your instances. In addition to the examples discussed below, there are a variety of tools that people have written that you can use to access your data in Amazon S3 from your computer or your instance. Some of the common ones are discussed in the AWS forums.

If you have permission, you can copy a file to or from Amazon S3 and your instance using one of the following methods.

GET or wget

The **wget** utility is an HTTP and FTP client that allows you to download public objects from Amazon S3. It is installed by default in Amazon Linux and most other distributions. To download an Amazon S3 object, use the following command, substituting the URL of the object to download.

```
wget http://s3.amazonaws.com/my_bucket/my_folder/my_file.ext
```

This method requires that the object you request is public; if the object is not public, you receive an `ERROR 403: Forbidden` message. If you receive this error, open the Amazon S3 console and change the permissions of the object to public. For more information, see the Amazon Simple Storage Service Developer Guide.

AWS Command Line Interface

The AWS Command Line Interface (AWS CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts. The AWS CLI allows users to authenticate themselves and download restricted items from Amazon S3 and also to upload items. For more information, such as how to install and configure the tools, see the AWS Command Line Interface detail page.

The **aws s3 cp** command is similar to the Unix **cp** command (the syntax is: **aws s3 cp** *source destination*). You can copy files from Amazon S3 to your instance, you can copy files from your instance to Amazon S3, and you can even copy files from one Amazon S3 location to another.

Use the following command to copy an object from Amazon S3 to your instance.

```
$ aws s3 cp s3://my_bucket/my_folder/my_file.ext my_copied_file.ext
```

Use the following command to copy an object from your instance back into Amazon S3.

```
$ aws s3 cp my_copied_file.ext s3://my_bucket/my_folder/my_file.ext
```

Use the following command to copy an object from one Amazon S3 location to another.

```
$ aws s3 cp s3://my_bucket/my_folder/my_file.ext s3://my_buck
et/my_folder/my_file2.ext
```

The **aws s3 sync** command can synchronize an entire Amazon S3 bucket to a local directory location. This can be helpful for downloading a data set and keeping the local copy up-to-date with the remote set. The command syntax is: **aws s3 sync** *source destination*. If you have the proper permissions on the Amazon S3 bucket, you can push your local directory back up to the cloud when you are finished by reversing the source and destination locations in the command.

Use the following command to download an entire Amazon S3 bucket to a local directory on your instance.

```
$ aws s3 sync s3://remote_S3_bucket local_directory
```

AWS Tools for Windows PowerShell

Windows instances have the benefit of a graphical browser that you can use to access the Amazon S3 console directly; however, for scripting purposes, Windows users can also use the AWS Tools for Windows PowerShell to move objects to and from Amazon S3.

Use the following command to copy an Amazon S3 object to your Windows instance.

```
PS C:\> Copy-S3Object -BucketName my_bucket -Key my_folder/my_file.ext -LocalFile
 my_copied_file.ext
```

Amazon S3 API

If you are a developer, you can use an API to access data in Amazon S3. For more information, see the Amazon Simple Storage Service Developer Guide. You can use this API and its examples to help develop your application and integrate it with other APIs and SDKs, such as the `boto` Python interface.

Block Device Mapping

Each Amazon EC2 instance that you launch has an associated root device volume, either an Amazon Elastic Block Store (Amazon EBS) volume or an instance store volume. You can use block device mapping to specify additional Amazon EBS volumes or instance store volumes to attach to an instance when it's launched. You can also attach additional Amazon EBS volumes to a running instance; see Attaching an Amazon EBS Volume to an Instance (p. 529). 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. 17).

Contents

Block Device Mapping Concepts

A *block device* is a storage device that moves data in sequences of bytes or bits (blocks). These devices support random access and generally use buffered I/O. Examples include hard disks, CD-ROM drives, and flash drives. A block device can be physically attached to a computer or accessed remotely as if it were physically attached to the computer. Amazon EC2 supports two types of block devices:

- Instance store volumes (virtual devices whose underlying hardware is physically attached to the host computer for the instance)
- Amazon EBS volumes (remote storage devices)

A *block device mapping* defines the block devices to be attached to an Amazon EC2 instance and the device name to use. You can specify a block device mapping as part of creating an AMI so that the mapping is used by all instances launched from the AMI. Alternatively, you can specify a block device mapping when you launch an instance, so this mapping overrides the one specified in the AMI from which you launched the instance.

There are two types of virtualization available in Amazon EC2: paravirtual (PV) and hardware virtual machine (HVM). The virtualization type is determined by the AMI used to launch the instance; some instance types support both PV and HVM while others support only one or the other. Be sure to note the virtualization type used by your AMI when you are creating your block device mapping because the recommended and available device names that you can use are different based on the virtualization type of your instance. For more information, see Virtualization Types (p. 58).

Specifying a Block Device Mapping

Use a block device mapping to attach instance store volumes and Amazon EBS volumes to an EC2 instance.

When you create a block device mapping, you specify this information for each block device that you need to attach to the instance:

- The device name within Amazon EC2, as shown in this table. The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different from the name that Amazon EC2 recommends.

Virtualization Type	Available	Reserved for Root	Used for Instance Store Volumes	Recommended for EBS Volumes
Paravirtual	/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]
HVM	/dev/sd[a-z] /dev/xvd[b-c][a-z]	Differs by AMI /dev/sda1 or /dev/xvda	/dev/sd[b-e] /dev/sd[b-y] (hs1.8xlarge)	/dev/sd[f-p]

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]`.

Hardware virtual machine (HVM) AMIs don't support the use of trailing numbers on device names (`xvd[a-p][1-15]`).

Depending on the block device driver of the kernel, the device might be attached with a different name than what you specify. For example, if you specify a device name of `/dev/sdh`, your device might be renamed `/dev/xvdh` or `/dev/hdh` by the kernel; in most cases, the trailing letter remains the same. In some versions of Red Hat Enterprise Linux (and its variants, such as CentOS), even the trailing letter might also change (where `/dev/sda` could become `/dev/xvde`). In these cases, each device name trailing letter is incremented the same number of times. For example, `/dev/sdb` would become `/dev/xvdf` and `/dev/sdc` would become `/dev/xvdg`. Amazon Linux AMIs create a symbolic link from the renamed device path to the name you specify, but other AMIs might behave differently.

- [Instance store volumes only] The virtual device: `ephemeral[0-3]`.
- [Amazon EBS volumes only] The ID of the snapshot to use to create the block device (snap-*xxxxxxxx*). This value is optional as long as you specify a volume size.
- [Amazon EBS volumes only] The size of the volume, in GiB. The specified size must be greater than or equal to the size of the specified snapshot.
- [Amazon EBS volumes only] Whether to delete the volume on instance termination (`true` or `false`). The default value is `true`.
- [Amazon EBS volumes only] The volume type, which can be `gp2` for General Purpose (SSD) volumes, `standard` for Magnetic volumes or `io1` for Provisioned IOPS (SSD) volumes. The default value is `standard` for Magnetic volumes.
- [Amazon EBS volumes only] The number of input/output operations per second (IOPS) that the volume supports. (Not used with General Purpose (SSD) or Magnetic volumes.)

Block Device Mapping Instance Store Caveats

There are several caveats to consider when launching instances with AMIs that have instance store volumes in their block device mappings.

- Some instance types include more instance store volumes than others, and some instance types contain no instance store volumes at all. If your instance type supports one instance store volume, and your AMI has mappings for two instance store volumes, then the instance launches with one instance store volume.
- Instance store volumes can only be mapped at launch time. You cannot stop an instance without instance store volumes (such as the `t2.micro`), change the instance to a type that supports instance store volumes, and then restart the instance with instance store volumes. However, you can create an AMI from the instance and launch it on an instance type that supports instance store volumes, and map those instance store volumes to the instance.
- If you launch an instance with instance store volumes mapped, and then stop the instance and change it to an instance type with fewer instance store volumes and restart it, the instance store volume mappings from the initial launch still show up in the instance metadata. However, only the maximum number of supported instance store volumes for that instance type are available to the instance.

    Note
    When an instance is stopped, all data on the instance store volumes is lost.

- Depending on instance store capacity at launch time, M3 instances may ignore AMI instance store block device mappings at launch unless they are specified at launch. You should specify instance store block device mappings at launch time, even if the AMI you are launching has the instance store volumes mapped in the AMI, to ensure that the instance store volumes are available when the instance launches.

Example Block Device Mapping

This figure shows an example block device mapping for an Amazon EBS–backed instance. It maps `/dev/sdb` to `ephemeral0` and maps two Amazon EBS volumes, one to `/dev/sdh` and the other to `/dev/sdj`. It also shows the Amazon EBS volume that is the root device volume, `/dev/sda1`.
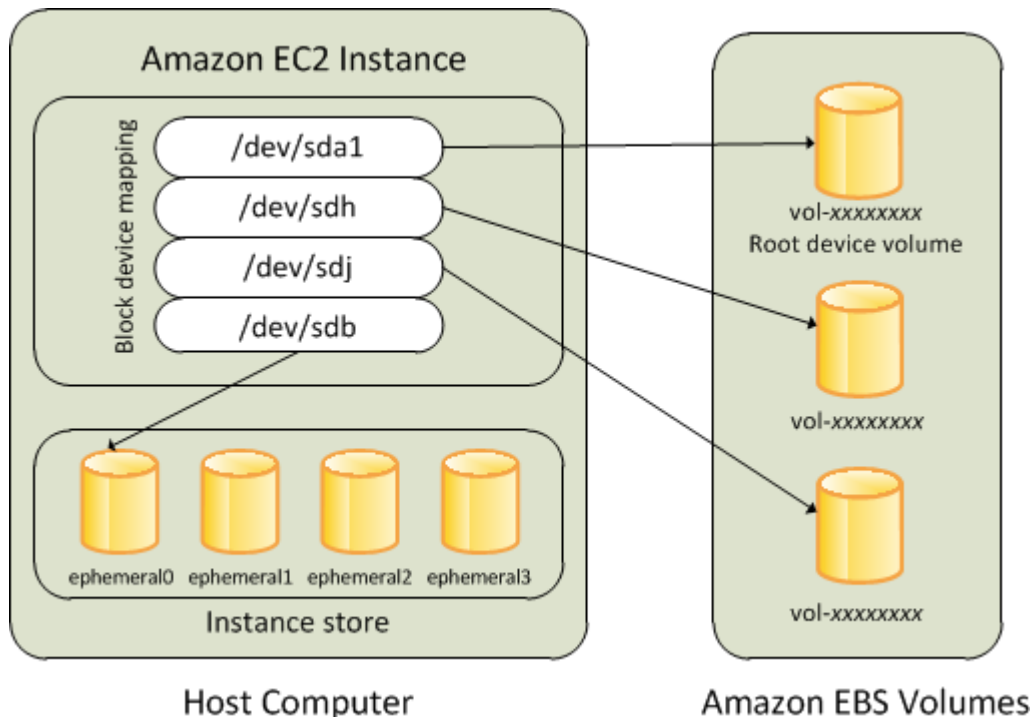
Note that this example block device mapping is used in the example commands and APIs in this topic. You can find example commands and APIs that create block device mappings here:

- Specifying a Block Device Mapping for an AMI (p. 595)
- Updating the Block Device Mapping when Launching an Instance (p. 598)

How Devices Are Made Available in the Operating System

Device names like `/dev/sdh` and `xvdh` are used by Amazon EC2 to describe block devices. The block device mapping is used by Amazon EC2 to specify the block devices to attach to an EC2 instance. After a block device is attached to an instance, it must be mounted by the operating system before you can access the storage device. When a block device is detached from an instance, it is unmounted by the operating system and you can no longer access the storage device.

With a 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. 581). The block device driver for the instance determines which devices are used when the volumes are formatted and mounted. For more information, see Attaching an Amazon EBS Volume to an Instance (p. 529).

Viewing Block Device Mappings

You can view information about each block device in a block device mapping. For details, see:

AMI Block Device Mapping

Each AMI has a block device mapping that specifies the block devices to attach to an instance when it is launched from the AMI. An AMI that Amazon provides includes a root device only. To add more block devices to an AMI, you must create your own AMI.

Contents

Specifying a Block Device Mapping for an AMI

There are two ways to specify volumes in addition to the root volume when you create an AMI. If you've already attached volumes to a running instance before you create an AMI from the instance, the block device mapping for the AMI includes those same volumes. For Amazon EBS volumes, the existing data is saved to a new snapshot, and it's this new snapshot that's specified in the block device mapping. For instance store volumes, the data is not preserved.

For an Amazon EBS-backed AMI, you can add Amazon EBS volumes and instance store volumes using a block device mapping. For an instance store-backed AMI, you can add only instance store volumes using a block device mapping.

> **Note**
> For M3 instances, you must specify instance store volumes in the block device mapping for the instance when you launch it. When you launch an M3 instance, instance store volumes specified in the block device mapping for the AMI may be ignored if they are not specified as part of the instance block device mapping.

To add volumes to an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**.
3. Select an instance, click **Actions**, and then select **Create Image**.
4. In the **Create Image** dialog box, click **Add New Volume**.
5. Select a volume type from the **Type** list and a device name from the **Device** list. For an Amazon EBS volume, you can optionally specify a snapshot, volume size, and volume type.
6. Click **Create Image**.

To add volumes to an AMI using the AWS CLI

Use the create-image command to specify a block device mapping for an Amazon EBS-backed AMI. Use the register-image command to specify a block device mapping for an instance store-backed AMI.

Specify the block device mapping using the following parameter:

```
--block-device-mappings [mapping, ...]
```

To add an instance store volume, use the following mapping:

```
{
    "DeviceName": "/dev/sdf",
    "VirtualName": "ephemeral0"
}
```

To add an empty 100 GiB Magnetic volume, use the following mapping:

```
{
    "DeviceName": "/dev/sdg",
    "Ebs": {
      "VolumeSize": 100
    }
}
```

To add an EBS volume based on a snapshot, use the following mapping:

```
{
    "DeviceName": "/dev/sdh",
    "Ebs": {
      "SnapshotId": "snap-xxxxxxxx"
    }
}
```

To omit a mapping for a device, use the following mapping:

```
{
    "DeviceName": "/dev/sdj",
    "NoDevice": ""
}
```

To add volumes to an AMI using the Amazon EC2 CLI

Use the ec2-create-image command to specify a block device mapping for an Amazon EBS-backed AMI.
Use the ec2-register command to specify a block device mapping for an instance store-backed AMI.

Specify the block device mapping using the following parameter:

```
-b "devicename=blockdevice"
```

devicename
    The device name within Amazon EC2
blockdevice
    To omit a mapping for the device from the AMI, specify `none`.

    To add an instance store volume, specify `ephemeral[0..3]`.

    To add an Amazon EBS volume to an Amazon EBS-backed instance, specify [*snapshot-id*]:[*size*]:[*delete-on-termination*]:[*type*[:*iops*]]

    • To add an empty volume, omit the snapshot ID and specify a volume size instead.

- To indicate whether the volume should be deleted on termination, specify `true` or `false`; the default value is `true`.
- To create a Provisioned IOPS (SSD) volume, specify `io1` and to create a General Purpose (SSD) volume, specify `gp2`; the default type is `standard` for Magnetic volumes. If the type is `io1`, you can also provision the number of IOPS the volume supports.

You can specify multiple block devices in a single command using multiple `-b` parameters. For example, the following parameters add an instance store volume as `/dev/sdb`, an Amazon EBS volume based on a snapshot as `/dev/sdh`, and an empty 100 GiB Amazon EBS volume as `/dev/sdj`.

```
-b "/dev/sdb=ephemeral0" -b "/dev/sdh=snap-d5eb27ab" -b "/dev/sdj=:100"
```

Viewing the Amazon EBS Volumes in an AMI Block Device Mapping

You can easily enumerate the Amazon EBS volumes in the block device mapping for an AMI.

To view the Amazon EBS volumes for an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. Select **EBS images** from the **Filter** drop-down list to get a list of Amazon EBS-backed AMIs.
4. Select the desired AMI, and look at the **Details** tab. At a minimum, the following information is available for the root device:

   - **Root Device Type** (`ebs`)
   - **Root Device Name** (for example, `/dev/sda1`)
   - **Block Devices** (for example, `/dev/sda1=snap-e1eb279f:8:true`)

   If the AMI was created with additional Amazon EBS volumes using a block device mapping, the **Block Devices** field displays the mapping for those additional volumes as well. (Recall that this screen doesn't display instance store volumes.)

To view the Amazon EBS volumes for an AMI using the AWS CLI

Use the describe-images command to enumerate the Amazon EBS volumes in the block device mapping for an AMI.

To view the Amazon EBS volumes for an AMI using the Amazon EC2 CLI

Use the ec2-describe-images command to enumerate the Amazon EBS volumes in the block device mapping for an AMI.

Instance Block Device Mapping

By default, an instance that you launch includes any storage devices specified in the block device mapping of the AMI from which you launched the instance. You can specify changes to the block device mapping for an instance when you launch it, and these updates overwrite or merge with the block device mapping of the AMI. However, you can only modify the volume size, volume type, and **Delete on Termination** flag on the block device mapping entry for the root device volume.

Contents

Updating the Block Device Mapping when Launching an Instance

You can add Amazon EBS volumes and instance store volumes to an instance when you launch it. Note that updating the block device mapping for an instance doesn't make a permanent change to the block device mapping of the AMI from which it was launched.

To add volumes to an instance using the console

1.   Open the Amazon EC2 console.
2.   From the Amazon EC2 console dashboard, click **Launch Instance**.
3.   On the **Choose an Amazon Machine Image (AMI)** page, choose the AMI to use and click **Select**.
4.   Follow the wizard to complete the **Choose an Instance Type** and **Configure Instance Details** pages.
5.   On the **Add Storage** page, you can modify the root volume, Amazon EBS volumes, and instance store volumes as follows:

     - To change the size of the root volume, locate the **Root** volume under the **Type** column, and change its **Size** field.
     - To suppress an Amazon EBS volume specified by the block device mapping of the AMI used to launch the instance, locate the volume and click its **Delete** icon.
     - To add an Amazon EBS volume, click **Add New Volume**, select **EBS** from the **Type** list, and fill in the fields (**Device**, **Snapshot**, and so on).
     - To suppress an instance store volume specified by the block device mapping of the AMI used to launch the instance, locate the volume, and click its **Delete** icon.
     - To add an instance store volume, click **Add New Volume**, select **Instance Store** from the **Type** list, and select a device name from **Device**.

6.   Complete the remaining wizard pages, and then click **Launch**.

To add volumes to an instance using the AWS CLI

Use the run-instances command to specify a block device mapping for an instance.

Specify the block device mapping using the following parameter:

```
--block-device-mappings [mapping, ...]
```

For example, suppose that an Amazon EBS-backed AMI specifies the following block device mapping:

- /dev/sdb=ephemeral0
- /dev/sdh=snap-92d333fb
- /dev/sdj=:100

To prevent `/dev/sdj` from attaching to an instance launched from this AMI, use the following mapping:

```
{
    "DeviceName": "/dev/sdj",
    "NoDevice": ""
}
```

To increase the size of `/dev/sdh` to 300 GiB, specify the following mapping. Notice that you don't need to specify the snapshot ID for `/dev/sdh`, because specifying the device name is enough to identify the volume.

```
{
    "DeviceName": "/dev/sdh",
    "Ebs": {
      "VolumeSize": 300
    }
}
```

To attach an additional instance store volume, `/dev/sdc`, specify the following mapping. If the instance type doesn't support multiple instance store volumes, this mapping has no effect.

```
{
    "DeviceName": "/dev/sdc",
    "VirtualName": "ephemeral1"
}
```

To add volumes to an instance using the Amazon EC2 CLI

Use the ec2-run-instances command to specify a block device mapping for an instance.

Specify the block device mapping using the following parameter:

```
-b "devicename=blockdevice"
```

devicename
    The device name within Amazon EC2
blockdevice
    To omit a mapping for the device from the AMI, specify `none`.

    To add an instance store volume, specify `ephemeral[0..3]`.

    To add an Amazon EBS volume to an EBS-backed instance, specify [*snapshot-id*]:[*size*]:[*delete-on-termination*]:[*type*[:*iops*]].
    - To add an empty Amazon EBS volume, omit the snapshot ID and specify a volume size instead.
    - To indicate whether the Amazon EBS volume is deleted on termination, specify `true` or `false`; the default value is `true`.
    - To create a Provisioned IOPS (SSD) volume, specify `io1` and to create a General Purpose (SSD) volume, specify `gp2`; the default type is `standard` for Magnetic volumes. If the type is `io1`, you can also provision the number of IOPS the volume supports.

For example, suppose that an EBS-backed AMI specifies the following block device mapping:

- /dev/sdb=ephemeral0
- /dev/sdh=snap-92d333fb
- /dev/sdj=:100

To prevent `/dev/sdj` from attaching to an instance launched from this AMI, use the following option:

```
-b "/dev/sdj=none"
```

To increase the size of `/dev/sdh` to 300 GiB, use the following option:

```
-b "/dev/sdh=:300"
```

Notice that you 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 the following option. If the instance type doesn't support multiple instance store volumes, this option has no effect.

```
-b "/dev/sdc=ephemeral1"
```

Viewing the Amazon EBS Volumes in an Instance Block Device Mapping

You can easily enumerate the Amazon EBS volumes mapped to an instance.

Note
For instances launched before the release of the 2009-10-31 API, AWS can't display the block device mapping. You must detach and reattach the volumes so that AWS can display the block device mapping.

To view the Amazon EBS volumes for an instance using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**.
3. In the search bar, select **Root Device Type**, and then select **EBS**. This displays a list of Amazon EBS–backed instances.
4. Locate and click the desired instance and look at the details displayed in the **Description** tab. At a minimum, the following information is available for the root device:

   - **Root device type** (`ebs`)
   - **Root device** (for example, `/dev/sda1`)
   - **Block devices** (for example, `/dev/sda1`, `/dev/sdh`, and `/dev/sdj`)

   If the instance was launched with additional Amazon EBS volumes using a block device mapping, the **Block devices** box displays those additional volumes as well as the root device. (Recall that this dialog box doesn't display instance store volumes.)

Root device type	ebs
Root device	/dev/sda1
Block devices	/dev/sda1
	/dev/sdf

5. To display additional information about a block device, click its entry next to **Block devices**. This displays the following information for the block device:

   - **EBS ID** (vol-*xxxxxxxx*)

- **Root device type** (ebs)
- **Attachment time** (*yyyy-mm*T*hh:mm:ss.ssTZD*)
- **Block device status** (`attaching`, `attached`, `detaching`, `detached`)
- **Delete on termination** (`Yes`, `No`)

To view the Amazon EBS volumes for an instance using the AWS CLI

Use the describe-instances command to enumerate the Amazon EBS volumes in the block device mapping for an instance.

To view the Amazon EBS volumes for an instance using the Amazon EC2 CLI

Use the ec2-describe-instances command to enumerate the Amazon EBS volumes in the block device mapping for an instance.

Viewing the Instance Block Device Mapping for Instance Store Volumes

When you view the block device mapping for your instance, you can see only the Amazon EBS volumes, not the instance store volumes. You can use instance metadata to query the complete block device mapping. The base URI for all requests for instance metadata is `http://169.254.169.254/latest/`.

First, connect to your running instance.

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 information, see Instance Metadata and User Data (p. 256).

Using Public Data Sets

Amazon Web Services provides a repository of public data sets that can be seamlessly integrated into AWS cloud-based applications. Amazon stores the data sets at no charge to the community and, as with all AWS services, you pay only for the compute and storage you use for your own applications.

Contents

Public Data Set Concepts

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

For more information, go to the Public Data Sets on AWS Page.

Available Public Data Sets

Public data sets are currently available in the following categories:

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

Finding Public Data Sets

Before you can use a public data set, you must locate the data set and determine which format the data set is hosted in. The data sets are available in two possible formats: Amazon EBS snapshots or Amazon S3 buckets.

To find a public data set and determine its format

1. Go to the Public Data Sets Page to see a listing of all available public data sets. You can also enter a search phrase on this page to query the available public data set listings.
2. Click the name of a data set to see its detail page.
3. On the data set detail page, look for a snapshot ID listing to identify an Amazon EBS formatted data set or an Amazon S3 URL.

Data sets that are in Amazon EBS snapshot format are used to create new Amazon EBS volumes that you attach to an Amazon EC2 instance. For more information, see Creating a Public Data Set Volume from a Snapshot (p. 603).

For data sets that are in Amazon S3 format, you can use the AWS SDKs or the HTTP query API to access the information, or you can use the AWS CLI to copy or synchronize the data to and from your instance. For more information, see Amazon S3 and Amazon EC2 (p. 590).

You can also use Amazon Elastic MapReduce to analyze and work with public data sets. For more information, see What is Amazon EMR?.

Creating a Public Data Set Volume from a Snapshot

To use a public data set that is in Amazon EBS snapshot format, you create a new volume, specifying the snapshot ID of the public data set. You can create your new volume using the AWS Management Console as follows. If you prefer, you can use the ec2-create-volume command instead.

To create a public data set volume from a snapshot

1. Open the Amazon EC2 console.
2. From the navigation bar, select the region that your data set snapshot is located in.

    Important
    Snapshot IDs are constrained to a single region, and you cannot create a volume from a snapshot that is located in another region. In addition, you can only attach an Amazon EBS volume to an instance in the same Availability Zone. For more information, see Resource Locations (p. 605).

    If you need to create this volume in a different region, you can copy the snapshot to your required region and then restore it to a volume in that region. For more information, see Copying an Amazon EBS Snapshot (p. 561).

3. In the navigation pane, click **Volumes**.
4. Above the upper pane, click **Create Volume**.
5. In the **Create Volume** dialog box, in the **Type** list, select **General Purpose (SSD)**, **Provisioned IOPS (SSD)**, or Magnetic. For more information, see Amazon EBS Volume Types (p. 523).
6. In the **Snapshot** field, start typing the ID or description of the snapshot for your data set. Select the snapshot from the list of suggested options.

    Note
    If the snapshot ID you are expecting to see does not appear, you may have a different region selected in the Amazon EC2 console. If the data set you identified in Finding Public Data Sets (p. 602) does not specify a region on its detail page, it is likely contained in the us-east-1 (N. Virginia) region.

7. In the **Size** field, enter the size of the volume (in GiB or TiB), or verify the that the default size of the snapshot is adequate.

    Note
    If you specify both a volume size and a snapshot ID, the size must be equal to or greater than the snapshot size. When you select a volume type and a snapshot ID, minimum and maximum sizes for the volume are shown next to the **Size** list.

8. For Provisioned IOPS (SSD) volumes, in the **IOPS** field, enter the maximum number of input/output operations per second (IOPS) that the volume can support.
9. In the **Availability Zone** list, select the Availability Zone in which to launch the instance.

    Important
    Amazon EBS volumes can only be attached to instances in the same Availability Zone.

10. Click **Yes, Create**.

Important
If you created a larger volume than the default size for that snapshot (by specifying a size in Step 7 (p. 603)), you need to extend the file system on the volume to take advantage of the extra space. For more information, see Expanding the Storage Space of a Volume (p. 545).

Attaching and Mounting the Public Data Set Volume

After you have created your new data set volume, you need to attach it to an Amazon EC2 instance to access the data (this instance must also be in the same Availability Zone as the new volume). For more information, see Attaching an Amazon EBS Volume to an Instance (p. 529).

After you have attached the volume to an instance, you need to mount the volume on the instance. For more information, see Making an Amazon EBS Volume Available for Use (p. 531).

Resources and Tags

Amazon EC2 enables you to manage your Amazon EC2 resources, such as images, instances, volumes, and snapshots. For more information, see the following documentation.

Topics

Resource Locations

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

Resource	Type	Description
AWS Account	Global	You can use the same AWS account in all regions.
Key Pairs	Global or Regional	You can use the key pairs that you create using Amazon EC2 only in the region where you created them. You can create and upload an RSA key pair that you can use in all regions. For more information, see Amazon EC2 Key Pairs (p. 413).
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 key pair name, is tied to its region and can be used only in the region where you created the resource. Although you can create resources with the same name in multiple regions, they aren't related to each other.
AMIs	Regional	An AMI is tied to the region where its files are located within Amazon S3. You can copy an AMI from one region to another. For more information, see Copying an AMI (p. 85).
Elastic IP Addresses	Regional	An Elastic IP address is tied to a region and can be associated only with an instance in the same region.
Security Groups	Regional	A security group is tied to a region and can be assigned only to instances in the same region. You can't enable an instance to communicate with an instance outside its region using security group rules. Traffic from an instance in another region is seen as WAN bandwidth.
EBS Snapshots	Regional	An EBS snapshot is tied to its region and can only be used to create volumes in the same region. You can copy a snapshot from one region to another. For more information, see Copying an Amazon EBS Snapshot (p. 561).
EBS Volumes	Availability Zone	An Amazon EBS volume is tied to its Availability Zone and can be attached only to instances in the same Availability Zone.
Instances	Availability Zone	An instance is tied to the Availability Zones in which you launched it. However, note that its instance ID is tied to the region.

Listing and Filtering Your Resources

Amazon EC2 provides different *resources* that you can use. These resources include images, instances, volumes, and snapshots. You can get a list of some types of resource using the Amazon EC2 console. You can get a list of each type of resource using its corresponding command or API action. If you have many resources, you can filter the results to include only the resources that match certain criteria.

Topics

- Advanced Search (p. 607)
- Listing Resources Using the Console (p. 608)
- Filtering Resources Using the Console (p. 608)
- Listing and Filtering Using the CLI and API (p. 609)

Advanced Search

Advanced search allows you to search using a combination of filters to achieve precise results. You can filter by keywords, user-defined tag keys, and predefined resource attributes. Advanced search is currently offered for the following resources:

- Instances
- Volumes
- Snapshots
- Elastic IP addresses
- Key pairs

The specific search types available are:

- **Search by keyword**

  To search by keyword, type or paste what you're looking for in the search box, and then press Enter. For example, to search for a specific instance, you can type the instance ID.

- **Search by fields**

  You can also search by fields, tags, and attributes associated with a resource. For example, to find all instances in the stopped state:

  1. In the search box, start typing `Instance State`. As you type, you'll see a list of suggested fields.
  2. Select **Instance State** from the list.
  3. Select **Stopped** from the list of suggested values.
  4. To further refine your list, click the search box for more search options.

- **Advanced search**

  You can create advanced queries by adding multiple filters. For example, you can search by tags and see instances for the Flying Mountain project running in the Production stack, and then search by attributes to see all t2.micro instances, or all instances in us-west-2a, or both.

- **Inverse search**

  You can search for resources that do not match a specified value. For example, to list all instances that are not terminated, search by the **Instance State** field, and prefix the Terminated value with an exclamation mark (!).

- **Partial search**

  When searching by field, you can also enter a partial string to find all resources that contain the string in that field. For example, search by **Instance Type**, and then type `t2` to find all t2.micro, t2.small or t2.medium instances.

- **Regular expression**

  Regular expressions are useful when you need to match the values in a field with a specific pattern. For example, search by the Name tag, and then type `^s.*` to see all instances with a Name tag that start with an 's'.

After you have the precise results of your search, you can bookmark the URL for easy reference. In situations where you have thousands of instances, filters and bookmarks can save you a great deal of time; you don't have to run searches repeatedly.

Listing Resources Using the Console

You can view the most common Amazon EC2 resource types using the console. To view additional resources, use the command line interface or the API actions.

To list EC2 resources using the console

1. Open the Amazon EC2 console.

2. In the navigation pane, click the option that corresponds to the resource, such as **AMIs** or **Instances**.

> **EC2 Dashboard**
> Events
> Tags
> Reports
> Limits
>
> ☐ 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
>
> ☐ AUTO SCALING
> Launch Configurations
> Auto Scaling Groups
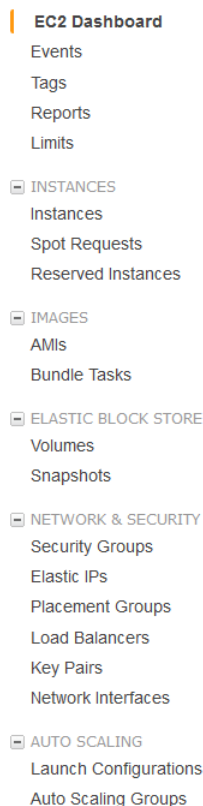
3. The page displays all the available resources.

Filtering Resources Using the Console

You can perform filtering and sorting of the most common resource types using the Amazon EC2 console. For example, you can use the search bar on the instances page to sort instances by tags, attributes, or keywords.

You can also use the search field on each page to find resources with specific attributes or values. You can use regular expressions to search on partial or multiple strings. For example, to find all instances that are using the MySG security group, enter `MySG` in the search field. The results will include any values that contain `MySG` as a part of the string, such as `MySG2` and `MySG3`. To limit your results to MySG only, enter `\bMySG\b` in the search field. To list all the instances whose type is either `m1.small` or `m1.large`, enter `m1.small|m1.large` in the search field.

> **Note**
> Not all of the screens provide the same search functionality.

To list volumes in the `us-east-1b` Availability Zone with a status of `available`

1. In the navigation pane, click **Volumes**.

2. Click on the search box, select **Attachment Status** from the menu, and then select **Detached**. (A detached volume is available to be attached to an instance in the same Availability Zone.)

3. Click on the search box again, select **State**, and then select **Available**.

4. Click on the search box again, select **Availability Zone**, and then select `us-east-1b`.

5. Any volumes that meet this criteria are displayed.

To list public 64-bit Linux AMIs backed by Amazon EBS

1. In the navigation pane, click **AMIs**.

2. In the **Filter** pane, select **Public images**, **EBS images**, and then your Linux distribution from the **Filter** lists.

3. Enter `x86_64` in the search field.

4. Any AMIs that meet this criteria are displayed.

Listing and Filtering Using the CLI and API

Each resource type has a corresponding CLI command or API request that you use to list resources of that type. For example, you can list Amazon Machine Images (AMI) using `ec2-describe-images` or `DescribeImages`. The response contains information for all your resources.

The resulting lists of resources can be long, so you might want to filter the results to include only the resources that match certain criteria. You can specify multiple filter values, and you can also specify multiple filters. For example, you can list all the instances whose type is either `m1.small` or `m1.large`, and that have an attached EBS volume that is set to delete when the instance terminates. The instance must match all your filters to be included in the results.

You can also use wildcards with the filter values. An asterisk (*) matches zero or more characters, and a question mark (?) matches exactly one character. For example, you can use `*database*` as a filter value to get all EBS snapshots that include `database` in the description. If you were to specify `database` as the filter value, then only snapshots whose description equals `database` would be returned. Filter values are case sensitive. We support only exact string matching, or substring matching (with wildcards).

> **Tip**
> Your search can include the literal values of the wildcard characters; you just need to escape them with a backslash before the character. For example, a value of `\*amazon\?\\` searches for the literal string `*amazon?\`.

For a list of supported filters per Amazon EC2 resource, see the relevant documentation:

- For the AWS CLI, see the relevant `describe` command in the *AWS Command Line Interface Reference*.
- For the Amazon EC2 CLI, see the relevant `ec2-describe` command in the *Amazon EC2 Command Line Reference*.
- For Windows PowerShell, see the relevant `Get` command in the *AWS Tools for Windows PowerShell Reference*.
- For the Query API, see the relevant `Describe` API action in the *Amazon EC2 API Reference*.

Tagging Your Amazon EC2 Resources

To help you manage your instances, images, and other Amazon EC2 resources, you can assign your own metadata to each resource in the form of *tags*. This topic describes tags and shows you how to create them.

Contents

Tag Basics

Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. Each tag consists of a key and an optional value, both of which you define. For example, you could define a set of tags for your account's Amazon EC2 instances that helps you track each instance's owner and stack level. We recommend that you devise a set of tag keys that meets your needs for each resource type. Using a consistent set of tag keys makes it easier for you to manage your resources. You can search and filter the resources based on the tags you add.

The following diagram illustrates how tagging works. In this example, you've assigned two tags to each of your instances, one called `Owner` and another called `Stack`. Each of the tags also has an associated value.
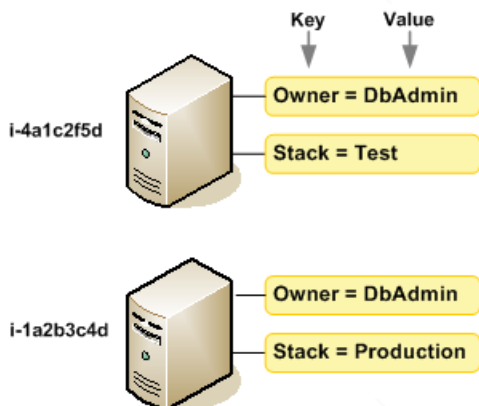
Tags don't have any semantic meaning to Amazon EC2 and are interpreted strictly as a string of characters. Also, tags are not automatically assigned to your resources.

You can work with tags using the AWS Management Console, the Amazon EC2 command line interface (CLI), and the Amazon EC2 API.

You can assign tags only to resources that already exist. When you use the Amazon EC2 console, you can access a list of tags to add to an instance, which will be applied immediately after the instance is created. If you add a tag that has the same key as an existing tag on that resource, the new value over-writes 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. 429).

Tag Restrictions

The following basic restrictions apply to tags:

- Maximum number of tags per resource—10
- Maximum key length—127 Unicode characters
- Maximum value length—255 Unicode characters
- Tag keys and values are case sensitive.
- Do not use the `aws:` prefix in your tag names or values because it is reserved for AWS use. You can't edit or delete tag names or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

You can't terminate, stop, or delete a resource based solely on its tags; you must specify the resource identifier. For example, to delete snapshots that you tagged with a tag key called `DeleteMe`, you must first get a list of those snapshots using `DescribeSnapshots` with a filter that specifies the tag. Then you use `DeleteSnapshots` with the resource identifiers of the snapshots (for example, snap-1a2b3c4d). You can't call `DeleteSnapshots` with a filter that specified the tag. For more information about using filters when listing your resources, see Listing and Filtering Your Resources (p. 606).

You can tag public or shared resources, but the tags you assign are available only to your AWS account and not to the other accounts sharing the resource.

You can't tag all resources, and some you can only tag using API actions or the command line. The following table lists all Amazon EC2 resources and the tagging restrictions that apply to them, if any. Resources with tagging restrictions of None can be tagged with API actions, the CLI, and the console.

Resource	Tagging support	Tagging restrictions
AMI	Yes	None
Bundle Task	No	
Customer Gateway	Yes	None
DHCP Option	Yes	None
EBS Volume	Yes	None
Instance Store Volume	No	
Elastic IP	No	
Instance	Yes	None
Internet Gateway	Yes	None
Key Pair	No	
Load Balancer	Yes	None
Network ACL	Yes	None
Network Interface	Yes	None
Placement Group	No	

Resource	Tagging support	Tagging restrictions
Reserved Instance	Yes	None
Reserved Instance Listing	No	
Route Table	Yes	None
Spot Instance Request	Yes	None
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
VPC Peering Connection	Yes	None
VPN Connection	Yes	None

For more information about tagging using the AWS console, see Working with Tags in the Console (p. 612). For more information about tagging using the API or command line, see API and CLI Overview (p. 617).

Tagging Your Resources for Billing

You can use tags to organize your AWS bill to reflect your own cost structure. To do this, sign up to get your AWS account bill with tag key values included. Then, to see the cost of your combined resources, you can organize your billing information based on resources that have the same tag key values. For example, you can tag several resources with a specific application name, and then organize your billing information to see the total cost of that application across several services. For more information, see Cost Allocation and Tagging in *About AWS Account Billing*.

Working with Tags in the Console

Using the Amazon EC2 console, you can see which tags are in use across all of your Amazon EC2 resources in the same region. You can view tags by resource and by resource type, and you can also view how many items of each resource type are associated with a specified tag. You can also use the Amazon EC2 console to apply or remove tags from one or more resources at a time.

Contents

- Displaying Tags (p. 612)
- Adding and Deleting Tags on an Individual Resource (p. 613)
- Adding and Deleting Tags to a Group of Resources (p. 614)
- Adding a Tag When You Launch an Instance (p. 616)
- Filtering a List of Resources by Tag (p. 616)

Displaying Tags

You can display tags in two different ways in the Amazon EC2 console. You can display the tags for an individual resource or for all resources.

To display tags for individual resources

When you select a resource-specific page in the Amazon EC2 console, it displays a list of those resources. For example, if you select **Instances** from the navigation pane, the console displays a list of Amazon EC2 instances. When you select a resource from one of these lists (e.g., an instance), if the resource supports tags, you can view and manage its tags. On most resource pages, you can view the tags in the **Tags** tab on the details pane. The following image shows the **Tags** tab for an instance with two tags: Name = DNS Server and Purpose = Network Management.

You can add a column to the resource list that displays all values for tags with the same key. This column enables you to sort and filter the resource list by the tag. There are two ways to add a new column to the resource list to display your tags.

- On the **Tags** tab, click **Show Column** for the tag.
- Click the **Show/Hide Columns** gear-shaped icon, and in the **Show/Hide Columns** dialog box, select the tag key under **Your Tag Keys**.

To display tags for all resources

You can display tags across all resources by selecting **Tags** from the navigation pane in the Amazon EC2 console. The following image shows the **Tags** pane, which lists all tags in use by resource type.

	Tag Key ▲	Tag Value ▾	Total ▾	Instances ▾	AMIs ▾	Volumes ▾
Manage Tag	Name	DNS Server	1	1	0	0
Manage Tag	Owner	TeamB	2	0	0	2
Manage Tag	Owner	TeamA	2	0	0	2
Manage Tag	Purpose	Project2	1	0	0	1
Manage Tag	Purpose	Logs	1	0	0	1
Manage Tag	Purpose	Network Management	1	1	0	0
Manage Tag	Purpose	Project1	2	0	0	2

Adding and Deleting Tags on an Individual Resource

You can manage tags for an individual resource directly from the resource's page. If you are managing an AMI's tags, the procedures are different from that of other resources. All procedures are explained below.

To add a tag to an individual resource

1. Open the Amazon EC2 console.
2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see Resource Locations (p. 605).
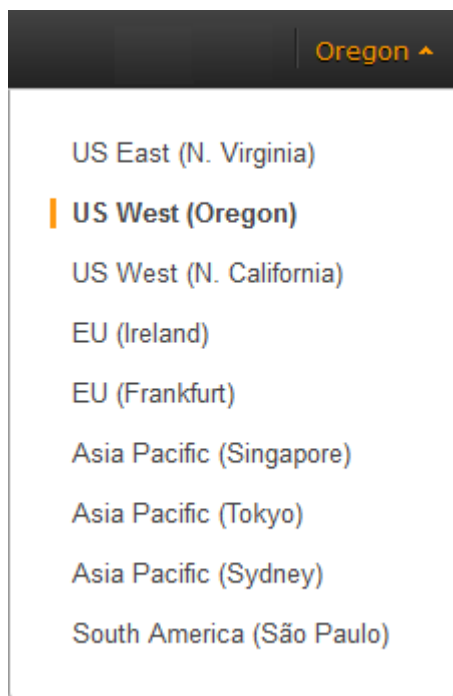
3. In the navigation pane, click a resource type (for example, **Instances**).

4. Select the resource from the resource list.

5. Select the **Tags** tab in the details pane.

6. Click the **Add/Edit Tags** button.

7. In the **Add/Edit Tags** dialog box, specify the key and value for each tag, and then click **Save**.

To delete a tag from an individual resource

1. Open the Amazon EC2 console.

2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see Resource Locations (p. 605).

3. In the navigation pane, click a resource type (for example, **Instances**).

4. Select the resource from the resource list.

5. Select the **Tags** tab in the details pane.

6. Click **Add/Edit Tags**, click the **Delete** icon for the tag, and click **Save**.

Adding and Deleting Tags to a Group of Resources

To add a tag to a group of resources

1. Open the Amazon EC2 console.

2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see Resource Locations (p. 605).
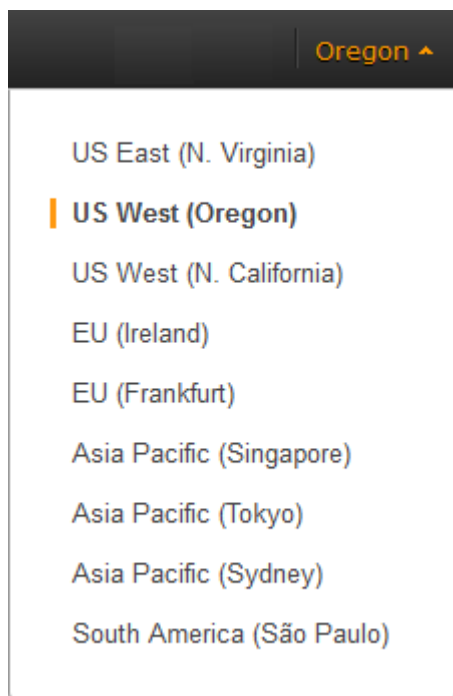
3. In the navigation pane, click **Tags**.
4. At the top of the content pane, click **Manage Tags**.
5. From the **Filter** drop-down list, select the type of resource (for example, instances) that you want to add tags to.
6. In the resources list, select the check box next to each resource that you want to add tags to.
7. In the **Key** and **Value** boxes under **Add Tag**, type the tag key and values you want, and then click **Add Tag**.

> **Note**
> If you add a new tag with the same tag key as an existing tag, the new tag overwrites the existing tag.

To remove a tag from a group of resources

1. Open the Amazon EC2 console.
2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see Resource Locations (p. 605).
3. In the navigation pane, click **Tags**.
4. At the top of the content pane, click **Manage Tags**.
5. To view the tags in use, click the **Show/Hide Columns** gear-shaped icon, and in the **Show/Hide Columns** dialog box, select the tag keys you want to view, and then click **Close**.
6. From the **Filter** drop-down list, select the type of resource (for example, instances) that you want to remove tags from.
7. In the resource list, select the check box next to each resource that you want to remove tags from.
8. Under **Remove Tag**, click in the **Key** box to select a key, or type its name, and then click **Remove Tag**.

Adding a Tag When You Launch an Instance

To add a tag using the Launch Wizard

1. From the navigation bar, select the region for the instance. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. Select the region that meets your needs. For more information, see Resource Locations (p. 605).
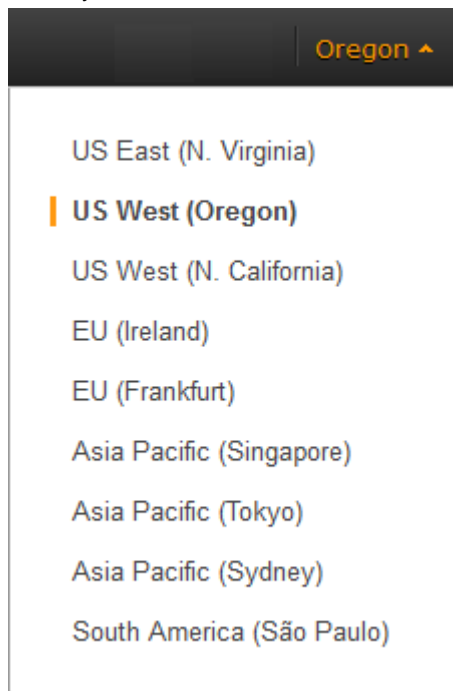
2. Click the **Launch Instance** button on the EC2 dashboard.
3. The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations called Amazon Machine Images (AMIs). Choose the AMI that you want to use and click its **Select** button. For more information about selecting an AMI, see Finding a Linux AMI (p. 58).
4. On the **Configure Instance Details** page, configure the instance settings as necessary, and then click **Next: Add Storage**.
5. On the **Add Storage** page, you can specify additional storage volumes for your instance. Click **Next: Tag Instance** when done.
6. On the **Tag Instance** page, specify tags for the instance by providing key and value combinations. Click **Create Tag** to add more than one tag to your instance. Click **Next: Configure Security Group** when you are done.
7. On the **Configure Security Group** page, you can choose from an existing security group that you own, or let the wizard create a new security group for you. Click **Review and Launch** when you are done.
8. Review your settings. When you're satisfied with your selections, click **Launch**. Select an existing key pair or create a new one, select the acknowledgment check box, and then click **Launch Instances**.

Filtering a List of Resources by Tag

You can filter your list of resources based on one or more tag keys and tag values.

To filter a list of resources by tag

1. Display a column for the tag as follows:

   a. Select one of the resources.

   b. Select the **Tags** tab in the details pane.

   c. Locate the tag in the list and click **Show Column**.

2. Click the filter icon in the top right corner of the column for the tag to display the filter list.

3. Select the tag values, and then click **Apply Filter** to filter the results list.

   Note
   Some screens have more advanced search functionality. For more information about this see Listing and Filtering Your Resources (p. 606).

API and CLI Overview

Use the following API and CLI commands to add, update, list, and delete the tags for your resources. The documentation for each command provides examples. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

Description	Amazon EC2 CLI	AWS CLI	AWS Tools for Windows PowerShell	API Action
Adds or overwrites one or more tags for the specified resource or resources.	ec2-create-tags	create-tags	New-EC2Tag	CreateTags
Deletes the specified tags from the specified resource or resources.	ec2-delete-tags	delete-tags	Remove-EC2Tag	DeleteTags
Describes one or more tags for your resources.	ec2-describe-tags	describe-tags	Get-EC2Tag	DescribeTags

You can also filter a list of resources according to their tags. For example syntax, see Filtering Resources Using the Console (p. 608). For a list of supported filters per Amazon EC2 resource, see the relevant documentation:

- For the AWS CLI, see the relevant `describe` command in the *AWS Command Line Interface Reference*.
- For the Amazon EC2 CLI, see the relevant `ec2-describe` command in the *Amazon EC2 Command Line Reference*.
- For Windows PowerShell, see the relevant `Get` command in the *AWS Tools for Windows PowerShell Reference*.
- For the Query API, see the relevant `Describe` API action in the *Amazon EC2 API Reference*.

Amazon EC2 Service Limits

Amazon EC2 provides different *resources* that you can use. These resources include images, instances, volumes, and snapshots. When you create your AWS account, we set default limits on these resources on a per-region basis. For example, there is a limit on the number of instances that you can launch in a region. Therefore, when you launch an instance in the US West (Oregon) region, the request must not cause your usage to exceed your current instance limit in that region.

The Amazon EC2 console provides limit information for the resources managed by the Amazon EC2 and Amazon VPC consoles. You can request an increase for many of these limits. Use the limit information that we provide to manage your AWS infrastructure. Plan to request any limit increases in advance of the time that you'll need them.

For more information about the limits for other services, see AWS Service Limits in the *Amazon Web Services General Reference*.

Viewing Your Current Limits

Use the **EC2 Service Limits** page in the Amazon EC2 console to view the current limits for resources provided by Amazon EC2 and Amazon VPC, on a per-region basis.

To view your current limits

1. Open the Amazon EC2 console.
2. From the navigation bar, select a region.
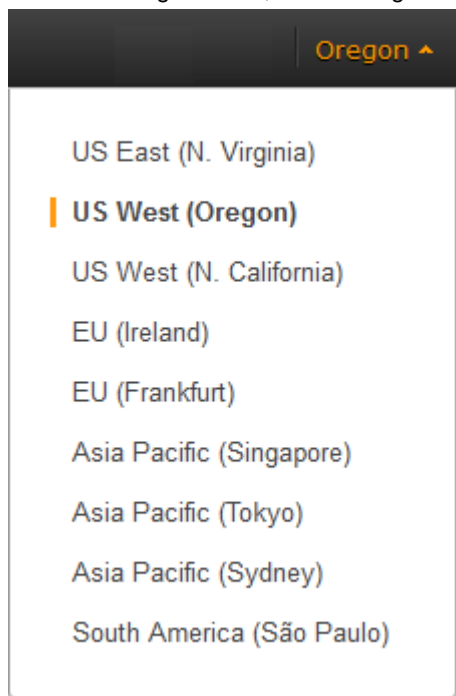
3. From the navigation pane, click **Limits**.
4. Locate the resource in the list. The **Current Limit** column displays the current maximum for that resource for your account.

Requesting a Limit Increase

Use the **Limits** page in the Amazon EC2 console to request an increase in the limits for resources provided by Amazon EC2 or Amazon VPC, on a per-region basis.

To request a limit increase

1. Open the Amazon EC2 console.
2. From the navigation bar, select a region.
3. From the navigation pane, click **Limits**.
4. Locate the resource in the list. Click **Request limit increase**.
5. Complete the required fields on the limit increase form. We'll respond to you using the contact method that you specified.

Amazon EC2 Usage Reports

The usage reports provided by Amazon EC2 enable you to analyze the usage of your instances in depth. The data in the usage reports is updated multiple times each day. You can filter the reports by AWS account, region, Availability Zone, operating system, instance type, purchasing option, tenancy, and tags.

To get usage and cost data for an account, you must have its account credentials and enable detailed billing reports with resources and tags for the account. If you're using consolidated billing and are logged into the payer account, you can view data for the payer account and all its linked accounts. If you're using consolidated billing and are logged into one of the linked accounts, you can only view data for that linked account. For information about consolidated billing, see Pay Bills for Multiple Accounts with Consolidated Billing.

Topics

- Available Reports (p. 619)
- Getting Set Up for Usage Reports (p. 619)
- Granting IAM Users Access to the Amazon EC2 Usage Reports (p. 621)
- Instance Usage Report (p. 621)
- Reserved Instance Utilization Reports (p. 625)

Available Reports

You can generate the following reports:

- Instance usage report (p. 621). This report covers your usage of on-demand instances, Spot Instances, and Reserved Instances.
- Reserved Instances utilization report (p. 625). This report covers the usage of your capacity reservation.

Getting Set Up for Usage Reports

Before you begin, enable detailed billing reports with resources and tags as shown in the following procedure. After you complete this procedure, we'll start collecting usage data for your instances. If you've already enabled detailed billing reports, you can access the usage data that we've been collecting since you enabled them.

Important
To complete these procedures, you must log in using your AWS account credentials. You can't complete these procedures if you log in using IAM user credentials.

To enable detailed billing reports

1. Select an existing Amazon S3 bucket to receive your usage data. Be sure to manage access to this bucket as it contains your billing data. (We don't require that you keep these files; in fact, you can delete them immediately if you don't need them.) If you don't have a bucket, create one as follows:

    a. Open the Amazon S3 console.

    b. Click **Create Bucket**.

    c. In the **Create a Bucket** dialog box, enter a name for your bucket (for example, *username*-ec2-usage-data), select a region, and then click **Create**. For more information about the requirements for bucket names, see Creating a Bucket in the *Amazon Simple Storage Service Console User Guide*.

2. Open the Billing and Cost Management console at https://console.aws.amazon.com/billing/home?#.

3. Click **Preferences** in the navigation pane.

4. Select **Receive Billing Reports**.

5. Specify the name of your Amazon S3 bucket in **Save to S3 Bucket**, and then click **Verify**.

6. Grant AWS permission to publish usage data to your Amazon S3 bucket.

    a. Under **Receive Billing Reports**, click **sample policy**. Copy the sample policy. Notice that the sample policy uses the bucket name you specified.

    b. Open the Amazon S3 console in another browser tab. Select your bucket, click **Properties**, and then expand **Permissions**. In the **Permissions** section, click **Add bucket policy**. Paste the sample policy into the text area and click **Save**. In the **Permissions** section, click **Save**.

    c. Return to the browser tab with the sample policy and click **Done**.

7. Under **Report**, select **Detailed billing report with resources and tags**.

8. Click **Save preferences**.

    Note
    It can take up to a day before you can see your data in the reports.

You can categorize your instances using tags. After you tag your instances, you must enable reporting on these tags.

To enable usage reporting by tag

1. Tag your instances. For best results, ensure that you add each tag you plan to use for reporting to each of your instances. For more information about how to tag an instance, see Tagging Your Amazon EC2 Resources (p. 610).

2. Open the Billing and Cost Management console at https://console.aws.amazon.com/billing/home?#.

3. Click **Preferences** in the navigation pane.

4. Under **Report**, click **Manage report tags**.

5. The page displays the list of tags that you've created. Select the tags that you'd like to use to filter or group your instance usage data, and then click **Save**. We automatically exclude any tags that you don't select from your instance usage report.

Note
We apply these changes only to the data for the current month. It can take up to a day for these changes to take effect.

Granting IAM Users Access to the Amazon EC2 Usage Reports

By default, IAM users can't access the Amazon EC2 usage reports. You must create an IAM policy that grants IAM users permission to access these reports.

The following policy allows users to view both Amazon EC2 usage reports.

```
{
  "Version": "2012-10-17",
  "Statement":[{
    "Effect": "Allow",
    "Action": "ec2-reports:*",
    "Resource": "*"
  }
  ]
}
```

The following policy allows users to view the instance usage report.

```
{
  "Version": "2012-10-17",
  "Statement":[{
    "Effect": "Allow",
    "Action": "ec2-reports:ViewInstanceUsageReport",
    "Resource": "*"
  }
  ]
}
```

The following policy allows users to view the Reserved Instances utilization report.

```
{
  "Version": "2012-10-17",
  "Statement":[{
    "Effect": "Allow",
    "Action": "ec2-reports:ViewReservedInstanceUtilizationReport",
    "Resource": "*"
  }
  ]
}
```

For more information, see Permissions and Policies in the *Using IAM* guide.

Instance Usage Report

You can use the instance usage report to view your instance usage and cost trends. You can see your usage data in either instance hours or cost. You can choose to see hourly, daily and monthly aggregates

of your usage data. You can filter or group the report by region, Availability Zone, instance type, AWS account, platform, tenancy, purchase option, or tag. After you configure a report, you can bookmark it so that it's easy to get back to later.

Here's an example of some of the questions that you can answer by creating an instance usage report:

- How much am I spending on instances of each instance type?
- How many instance hours are being used by a particular department?
- How is my instance usage distributed across Availability Zones?
- How is my instance usage distributed across AWS accounts?

Topics

Report Formats

We display the usage data that you request as both a graph and a table.

For example, the following graph displays cost by instance type. The key for the graph indicates which color represents which instance type. To get detailed information about a segment of a bar, hover over it.
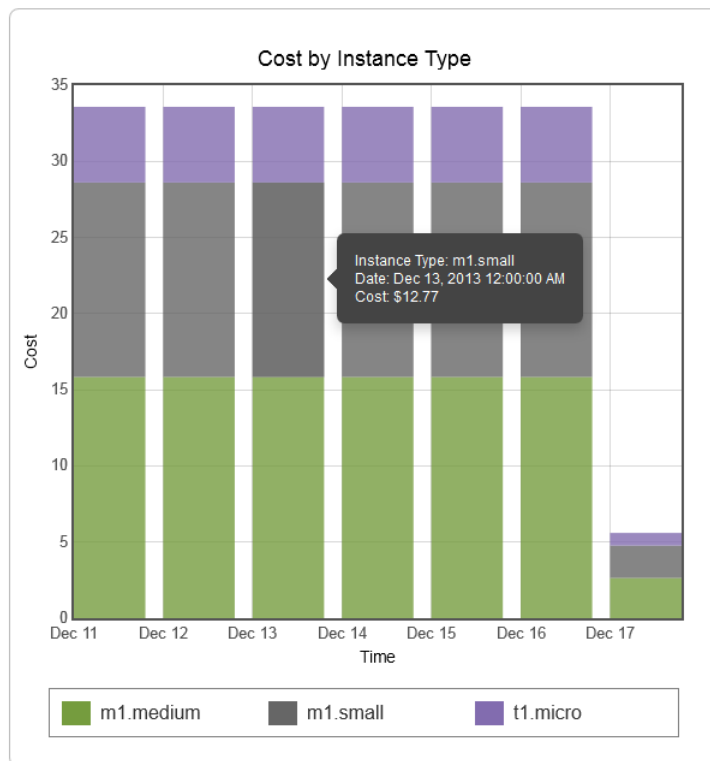
The corresponding table displays one column for each instance type. Notice that we include a color band in the column head that is the same color as the instance type in the graph.

Time (UTC)	m1.medium	m1.small	t1.micro
12/11/13	$15.84	$12.77	$4.97
12/12/13	$15.84	$12.77	$4.97
12/13/13	$15.84	$12.77	$4.97
12/14/13	$15.84	$12.77	$4.97
12/15/13	$15.84	$12.77	$4.97
12/16/13	$15.84	$12.77	$4.97
12/17/13	$2.64	$2.13	$0.83
Total	$97.68	$78.75	$30.65

Viewing Your Instance Usage

The following procedures demonstrate how to generate usage reports using some of the capabilities we provide.

Before you begin, you must get set up. For more information, see Getting Set Up for Usage Reports (p. 619).

To filter and group your instance usage by instance type

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Reports** and then click **EC2 Instance Usage Report**.
3. Select an option for **Unit**. To view the time that your instances have been running, in hours, select `Instance Hours`. To view the cost of your instance usage, select `Cost`.
4. Select options for **Granularity** and **Time range**.

   - To view the data summarized for each hour in the time range, select `Hourly` granularity. You can select a time range of up to 2 days when viewing hourly data.
   - To view the data summarized for each day in the time range, select `Daily` granularity. You can select a time range of up to 2 months when viewing daily data.
   - To view the data summarized for each month in the time range, select `Monthly` granularity.

5. In the **Filter** list, select `Instance Type`. In the **Group by** list, select `Instance Type`.
6. In the filter area, select one or more instance types and then click **Update Report**. The filters you specify appear under **Applied Filters**.

Notice that you can return to the Amazon EC2 console by clicking either **Reports** or **EC2 Management Console** at the top of the page.

To group your instance usage based on tags

1. Open the Instance Usage Reports page.
2. Select an option for **Unit**. To view the time that your instances have been running, in hours, select `Instance Hours`. To view the cost of your instance usage, select `Cost`.
3. Select options for **Granularity** and **Time range**.

   - To view the data summarized for each hour in the time range, select `Hourly` granularity. You can select a time range of up to 2 days when viewing hourly data.
   - To view the data summarized for each day in the time range, select `Daily` granularity. You can select a time range of up to 2 months when viewing daily data.
   - To view the data summarized for each month in the time range, select `Monthly` granularity.

4. In the **Group by** list, select **Tag**.
5. Click the **Key Name** box, select a name from the list, and then click **Update Report**. If there are no items in this list, you must enable usage reporting by tag. For more information, see To enable usage reporting by tag (p. 620).

Bookmarking a Customized Report

You might want to generate a customized report again. Do this by bookmarking the report.

To bookmark a custom report

1. Select the options and filters for your report. Each selection you make adds a parameter to the console URL. For example, `granularity=Hourly` and `Filters=`*filter_list*.
2. Using your browser, add the console URL as a bookmark.
3. To generate the same report in the future, use the bookmark that you created.

Exporting Your Usage Data

You might want to include your report graph or table in other reports. Do this by exporting the data.

To export usage data

1. Select the options and filters for your report.
2. To export the usage data from the table as a `.csv` file, click **Download** and select **CSV Only**.
3. To export the graphical usage data as a `.png` file, click **Download** and select **Graph Only**.

Reserved Instance Utilization Reports

The Reserved Instance utilization report describes the utilization over time of each group (or *bucket*) of Amazon EC2 Reserved Instances that you own. Each bucket has a unique combination of region, Availability Zone, instance type, tenancy, offering type, and platform. You can specify the time range that the report covers, from a single day to weeks, months, a year, or three years. The available data depends on when you enable detailed billing reports for the account (see Getting Set Up for Usage Reports (p. 619)). The Reserved Instance utilization report compares the Reserved Instance prices paid for instance usage in the bucket with On-Demand prices and shows your savings for the time range covered by the report.

Note
The Reserved Instance buckets aggregate Reserved Instances across Amazon VPC and non-Amazon VPC (EC2 Classic) network platform types in the same way that your bill is calculated. Additionally, Reserved Instances in a bucket may have different upfront and hourly prices.

Here are examples of some of the questions that you can answer using the Reserved Instance utilization report:

- How well am I utilizing my Reserved Instances?
- Are my Reserved Instances helping me save money?

For information about Reserved Instances, see Reserved Instances (p. 185).

Before you begin, you must get set up. For more information, see Getting Set Up for Usage Reports (p. 619).

Topics

Getting to Know the Report

The Reserved Instance utilization report displays your requested utilization data in graph and table formats.

The report aggregates Reserved Instance usage data for a given period by bucket. In the report, each row in the table represents a bucket and provides the following metrics:

- **Count**—The highest number of Reserved Instances owned at the same time during the period of the report.
- **Usage Cost**—The total Reserved Instance usage fees applied to instance usage covered by the Reserved Instance bucket.
- **Total Cost**—The usage cost plus the amortized upfront fee for the usage period associated with the Reserved Instance bucket.

    Note
    If the bucket contains a Reserved Instance that you sold in the Reserved Instances Marketplace and that Reserved Instance was active at any point during the period of the report, the total cost of the bucket might be inflated and your savings might be underestimated.

- **Savings**—The difference between what your usage for the period would have cost at On-Demand prices and what it actually cost using Reserved Instances (Total Cost).
- **Average Utilization**—The average hourly utilization rate for the Reserved Instance bucket over the period.
- **Maximum Utilization**—The highest utilization rate of any hour during the period covered by the report.

For each row—or Reserved Instance bucket—in the table, the graph represents data based on your selected **Show** metric over the selected **Time range** for the report. Each point in the graph represents a metric at a point in time. For information about report options, see Options Reference (p. 629).

A color band at the edge of each selected row in the table corresponds to a report line in the graph. You can show a row in the graph by selecting the checkbox at the beginning of the row.

	Instance Type	Platform	Tenancy	Availability Zone	Offering Type	Count	Usage Cost	Total Cost	Savings	Average Utilization	Maximum Utilization
☑	m1.large	Linux/Unix	Shared	eu-west-1a	Medium	12	$438.30	$491.04	$648.54	27.00%	100.00%

By default, the Reserved Instance utilization report returns data over the last 14 days for all Reserved Instance buckets. The graph shows the average utilization for the first five buckets in the table. You can customize the report graph to show different utilization (average utilization, maximum utilization) or cost (total cost, usage cost) data over a period ranging from a day to weeks, months, or years.

For example, the following report table shows Reserved Instance utilization for a two-month period. In this case, the period is December 1 through January 31. The report has been filtered to only return data about Reserved Instance buckets in eu-west-1a, eu-west-1b, and eu-west-1c.

	Instance Type	Platform	Tenancy	Availability Zone	Offering Type	Count	Usage Cost	Total Cost	Savings	Average Utilization	Maximum Utilization
☑	m1.large	Linux/Unix	Shared	eu-west-1a	Medium	12	$438.30	$491.04	$648.54	27.00%	100.00%
☑	m1.medium	Linux/Unix	Shared	eu-west-1a	Medium	2	$137.90	$164.27	$194.27	100.00%	100.00%
☑	m1.xlarge	Linux/Unix	Shared	eu-west-1b	Heavy	3	$665.28	$791.86	$1,359.38	100.00%	100.00%
☑	m1.small	Linux/Unix	Shared	eu-west-1a	Heavy	2	$55.44	$71.26	$108.01	100.00%	100.00%
☑	m1.large	Linux/Unix	Shared	eu-west-1b	Heavy	15	$1,663.20	$1,726.49	$3,417.87	95.00%	100.00%
☐	m1.small	Linux/Unix	Shared	eu-west-1a	Medium	10	$253.70	$293.25	$366.37	74.00%	100.00%
☐	m1.medium	Linux/Unix	Shared	eu-west-1a	Heavy	4	$151.76	$172.96	$387.08	100.00%	100.00%
☐	m1.large	Linux/Unix	Shared	eu-west-1a	Heavy	17	$1,884.96	$1,948.25	$3,663.33	92.00%	100.00%

The graph shows the daily Reserved Instance average utilization for the selected buckets.
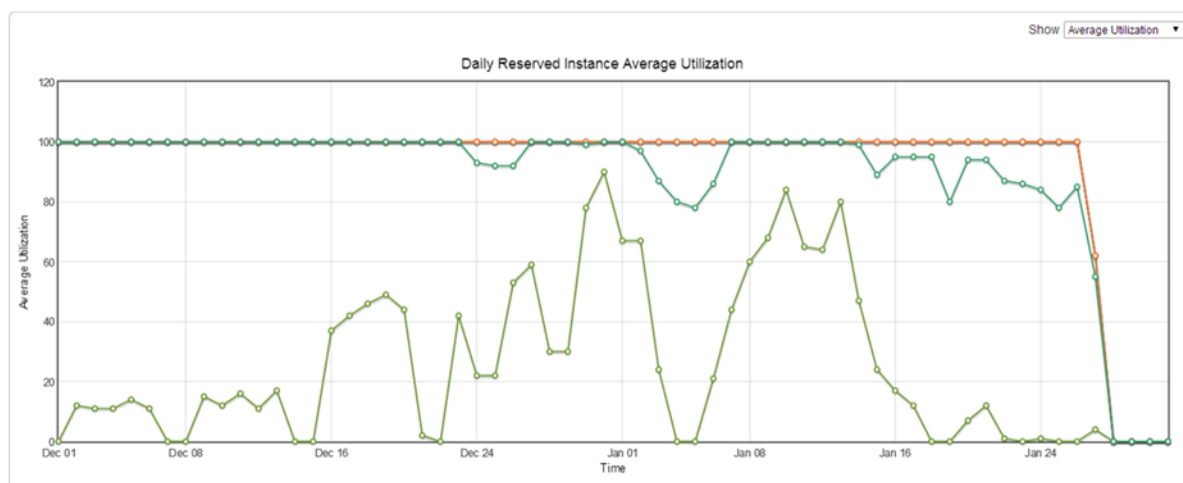
Customizing the Report

You can customize the Reserved Instance utilization report with **Time range** and **Filter** options.
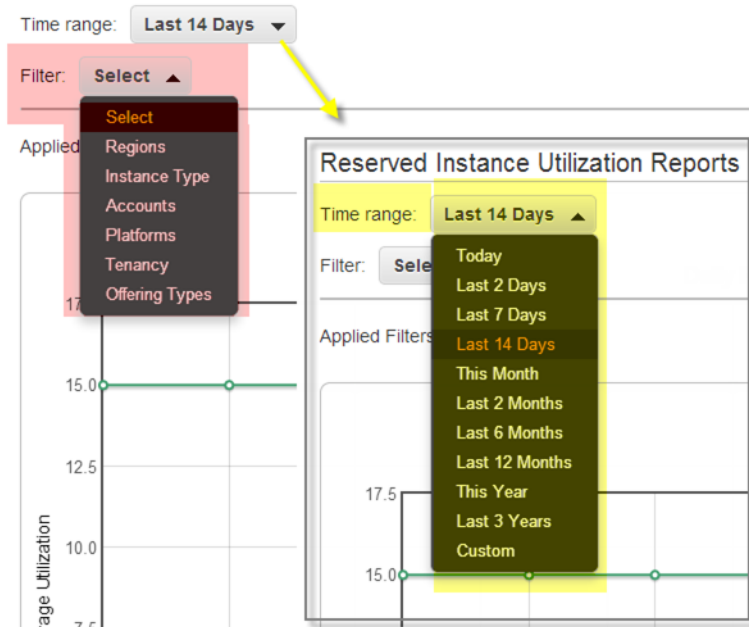
Time range provides a list of common relative time ranges, ranging from **Today** to **Last 3 Years**. Select the time range that works best for your needs, and then click **Update Report** to apply the change. To apply a time range that is not on the list, select **Custom** and enter the start date and end date for which you want to run the report.

Filter lets you scope your Reserved Instance utilization report by one or more of the following Reserved Instance qualities: region, instance type, account, platform, tenancy, and offering type. For example, you can filter by region or by specific Availability Zones in a region, or both. To filter by region, select **Regions**, then select the regions and Availability Zones you want to include in the report, and click **Update Report**.

The report will return all results if no filter is applied.

For information about report options, see .

Bookmarking a Customized Report

You might want to generate a customized report again. Do this by bookmarking the report.

To bookmark a custom report

1. Select the options and filters for your report. Each selection you make adds a parameter to the console URL. For example, `granularity=Hourly` and `Filters=`*filter_list*.
2. Using your browser, add the console URL as a bookmark.
3. To generate the same report in the future, use the bookmark that you created.

Exporting Your Usage Data

You might want to include your report graph or table in other reports. Do this by exporting the data.

To export usage data

1. Select the options and filters for your report.
2. To export the usage data from the table as a `.csv` file, click **Download** and select **CSV Only**.
3. To export the graphical usage data as a `.png` file, click **Download** and select **Graph Only**.

Options Reference

Use the **Show** options to specify the metric to be displayed by the report graph.

*   Average Utilization

    Shows the average of the utilization rates for each hour over the selected time range, where the utilization rate of a bucket for an hour is the number of instance hours used for that hour divided by the total number of Reserved Instances owned in that hour.
*   Maximum Utilization

    Shows the highest of the utilization rates of any hour over the selected time range, where the utilization rate of a bucket for an hour is the number of instance hours used for that hour divided by the total number of Reserved Instances owned in that hour.
*   Usage Cost

    Shows the total cost based on hourly fees for a selected bucket of Reserved Instances. For heavy utilization Reserved Instance buckets, usage cost is calculated by multiplying the number of hours in the time frame by the hourly rates associated with the Reserved Instances in the bucket. For medium and light utilization Reserved instances, usage cost is calculated by multiplying the number of instance hours used for the selected time frame by the hourly rates for the respective Reserved Instances in the bucket.
*   Total Cost

    Shows the usage cost plus the amortized portion of the upfront cost of the Reserved Instances in the bucket over the period for which the report is generated.

Use **Time range** to specify the period on which the report will be based.

> **Note**
> All times are specified in UTC time.

*   Today

Shows data for usage that takes place during any of the hours for the current calendar day. Can be used with hourly, daily, or monthly granularities.

- Last 2 Days

Shows data for usage that took place during the current and previous one calendar day. Can be used with hourly, daily, or monthly granularities.

- Last 7 Days

Shows data for usage that took place during the current and previous six calendar days. Can be used with daily or monthly granularities.

- Last 14 Days

Shows data for usage that took place during the current and previous 13 calendar days. Can be used with daily or monthly granularities.

- This Month

Shows data for usage that took place during the current calendar month. Can be used with daily or monthly granularities.

- Last 2 Months

Shows data for usage that took place during the current and previous calendar months. Can be used with daily or monthly granularities.

- Last 6 Months

Shows data for usage that took place during the current and previous five calendar months. Can be used with monthly granularities.

- Last 12 Months

Shows data for usage that took place during the current and previous 11 calendar months. Can be used with monthly granularity.

- This Year

Shows data for usage that took place during the current calendar year. Can be used with monthly granularity.

- Last 3 Years

Shows data for usage that took place during the current and previous two calendar years. Can be used with monthly granularity.

- Custom

Shows data for the time range for the entered **Start** and **End** dates specified in the following format: mm/dd/yyyy. Can be used with hourly, daily, or monthly granularities, but you can only specify a maximum time range of two days for hourly data, two months for daily data, and three years for monthly data.

Use **Filter** to scope the data displayed in the report.

- Regions
- Instance Types
- Accounts
- Platforms
- Tenancy
- Offering Type

Troubleshooting Instances

The following documentation can help you troubleshoot problems that you might have with your instance.

Topics

For additional help with Windows instances, see Troubleshooting Windows Instances in the *Amazon EC2 User Guide for Microsoft Windows Instances*.

You can also search for answers and post questions on the Amazon EC2 forum.

What To Do If An Instance Immediately Terminates

After you launch an instance, we recommend that you check its status to confirm that it goes from the `pending` status to the `running` status, the not `terminated` status.

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.

Getting the Reason for Instance Termination

You can use the Amazon EC2 console, CLI, or API to get information about the reason that the instance terminated.

To get the reason that an instance terminated using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances** to display the instance details.
3. Select your instance.
4. In the **Description** tab, locate the reason next to the label **State transition reason**. If the instance is still running, there's typically no reason listed. If you've explicitly stopped or terminated the instance, the reason is `User initiated shutdown`.

To get the reason that an instance terminated using the command line

1. Use the ec2-describe-instances command in verbose mode as follows:

```
$ 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>
```

This example response shows the reason code that you'll see after you stop or terminate a running instance. If the instance terminated immediately, you'll see `code` and `message` elements that describe the reason that the instance terminated (for example, `VolumeLimitExceeded`).

Troubleshooting Connecting to Your Instance

The following are possible problems you may have and error messages you may see while trying to connect to your instance.

Topics

For additional help with Windows instances, see Troubleshooting Windows Instances in the *Amazon EC2 User Guide for Microsoft Windows Instances*.

You can also search for answers and post questions on the Amazon EC2 forum.

Error connecting to your instance: Connection timed out

If you try to connect to your instance and get an error message `Network error: Connection timed out` or `Error connecting to [instance], reason: -> Connection timed out: connect`, try the following:

- Check your security group rules. You need a security group rule that allows inbound traffic to the proper port(s) from your public IP address. When you use SSH to connect to your Linux instance, this is usually port 22. When you use RDP to connect to your Windows instance, this is usually port 3389. For more information, see Authorizing Network Access to Your Instances (p. 467).
- [EC2-VPC] Check the route table for the subnet. You need a route that sends all traffic destined outside the VPC (`0.0.0.0/0`) to the Internet gateway for the VPC.
- Check the CPU load on your instance; the server may be overloaded. AWS automatically provides data such as Amazon CloudWatch metrics and instance status, which you can use to see how much CPU load is on your instance and, if necessary, adjust how your loads are handled. For more information, see Monitoring Your Instances with CloudWatch (p. 354).
  - If your load is variable, you can automatically scale your instances up or down using Auto Scaling and Elastic Load Balancing.
  - If your load is steadily growing, you can move to a larger instance type. For more information, see Resizing Your Instance (p. 128).
- Verify that you are using the private key file that corresponds to the key pair that was selected when the instance was launched. For more information, see Amazon EC2 Key Pairs (p. 413).
- Verify that you are connecting with the appropriate user name for your AMI.
  - For an Amazon Linux AMI, the user name is `ec2-user`.
  - For a RHEL5 AMI, the user name is either `root` or `ec2-user`.
  - For an Ubuntu AMI, the user name is `ubuntu`.
  - For a Fedora AMI, the user name is either `fedora` or `ec2-user`.
  - For SUSE Linux, the user name is `root`.
  - Otherwise, if `ec2-user` and `root` don't work, check with the AMI provider.

  If you are using MindTerm to connect, enter the user name in the **User name** box in the **Connect To Your Instance** window.

  If you are using PuTTY to connect, enter the user name in the **Host name** box in the **PuTTY Configuration** window.
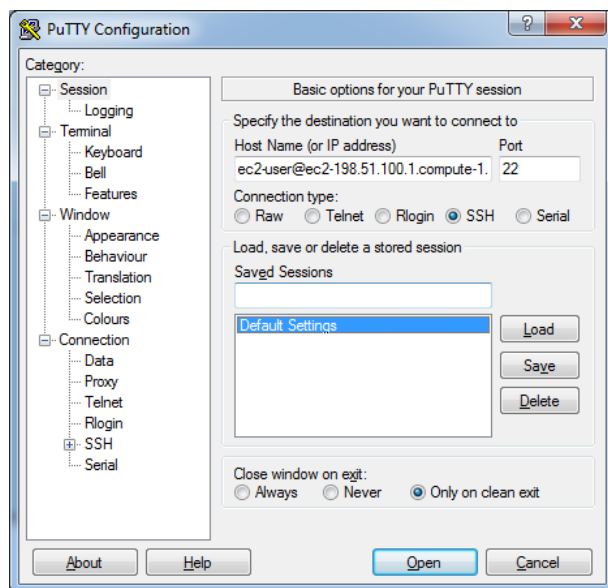
Error: User key not recognized by server

If you use SSH to connect to your instance

- Use `ssh -vvv` to get triple verbose debugging information while connecting:

```
#ssh -vvv -i [your key name].pem ec2-user@[public DNS address of your in
stance].compute-1.amazonaws.com
```

The following sample output demonstrates what you might see if you were trying to connect to your instance with a key that was not recognized by the server:

```
open/ANT/myusername/.ssh/known_hosts).
debug2: bits set: 504/1024
debug1: ssh_rsa_verify: signature correct
debug2: kex_derive_keys
debug2: set_newkeys: mode 1
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug2: set_newkeys: mode 0
debug1: SSH2_MSG_NEWKEYS received
debug1: Roaming not allowed by server
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug2: service_accept: ssh-userauth
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug2: key: boguspem.pem ((nil))
debug1: Authentications that can continue: publickey
debug3: start over, passed a different list publickey
debug3: preferred gssapi-keyex,gssapi-with-mic,publickey,keyboard-interact
ive,password
debug3: authmethod_lookup publickey
debug3: remaining preferred: keyboard-interactive,password
debug3: authmethod_is_enabled publickey
debug1: Next authentication method: publickey
```

Amazon Elastic Compute Cloud User Guide for Linux
Error: Host key not found, Permission denied (pub-
lickey), *or* Authentication failed, permission denied

```
debug1: Trying private key: boguspem.pem
debug1: read PEM private key done: type RSA
debug3: sign_and_send_pubkey: RSA
9c:4c:bc:0c:d0:5c:c7:92:6c:8e:9b:16:e4:43:d8:b2
debug2: we sent a publickey packet, wait for reply
debug1: Authentications that can continue: publickey
debug2: we did not send a packet, disable method
debug1: No more authentication methods to try.
Permission denied (publickey).
```

If you use SSH (MindTerm) to connect to your instance

- If Java is not enabled, the server does not recognize the user key. To enable Java, see How do I enable Java in my web browser? in the Java documentation.

If you use PuTTY to connect to your instance

- Verify that your private key (.pem) file has been converted to the format recognized by PuTTY (.ppk). For more information about converting your private key, see Connecting to Your Linux Instance from Windows Using PuTTY (p. 303).

  Note
  In PuTTYgen, load your private key file and select **Save Private Key** rather than **Generate**.

- Verify that you are connecting with the appropriate user name for your AMI. Enter the user name in the **Host name** box in the **PuTTY Configuration** window.

  - For an Amazon Linux AMI, the user name is `ec2-user`.

  - For a RHEL5 AMI, the user name is either `root` or `ec2-user`.

  - For an Ubuntu AMI, the user name is `ubuntu`.

  - For a Fedora AMI, the user name is either `fedora` or `ec2-user`.

  - For SUSE Linux, the user name is `root`.

  - Otherwise, if `ec2-user` and `root` don't work, check with the AMI provider.

- Verify that you have an inbound security group rule to allow inbound traffic to the appropriate port. For more information, see Authorizing Network Access to Your Instances (p. 467).

Error: Host key not found, Permission denied (publickey), *or* Authentication failed, permission denied

If you connect to your instance using SSH and get any of the following errors, `Host key not found in [directory]`, `Permission denied (publickey)`, or `Authentication failed, permission denied`, verify that you are connecting with the appropriate user name for your AMI *and* that you have specified the proper private key (`.pem`) file for your instance. For MindTerm clients, enter the user name in the **User name** box in the **Connect To Your Instance** window.

The appropriate user names are as follows:

- For an Amazon Linux AMI, the user name is `ec2-user`.

- For a RHEL5 AMI, the user name is either `root` or `ec2-user`.

- For an Ubuntu AMI, the user name is `ubuntu`.

- For a Fedora AMI, the user name is either `fedora` or `ec2-user`.

- For SUSE Linux, the user name is `root`.

- Otherwise, if `ec2-user` and `root` don't work, check with the AMI provider.

Error: Unprotected Private Key File

Your private key file must be protected from read and write operations from any other users. If your private key can be read or written to by anyone but you, then SSH ignores your key and you see the following warning message below.

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@           WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0777 for '.ssh/my_private_key.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
bad permissions: ignore key: .ssh/my_private_key.pem
Permission denied (publickey).
```

If you see a similar message when you try to log in to your instance, examine the first line of the error message to verify that you are using the correct public key for your instance. The above example uses the private key `.ssh/my_private_key.pem` with file permissions of `0777`, which allow anyone to read or write to this file. This permission level is very insecure, and so SSH ignores this key. To fix the error, execute the following command, substituting the path for your private key file.

```
$ chmod 0400 .ssh/my_private_key.pem
```

Error: Server refused our key *or* No supported authentication methods available

If you use PuTTY to connect to your instance and get either of the following errors, `Error: Server refused our key` or `Error: No supported authentication methods available`, verify that you are connecting with the appropriate user name for your AMI. Enter the user name in the **User name** box in the **PuTTY Configuration** window.

The appropriate user names are as follows:

- For an Amazon Linux AMI, the user name is `ec2-user`.

- For a RHEL5 AMI, the user name is either `root` or `ec2-user`.

- For an Ubuntu AMI, the user name is `ubuntu`.

- For a Fedora AMI, the user name is either `fedora` or `ec2-user`.

- For SUSE Linux, the user name is `root`.

- Otherwise, if `ec2-user` and `root` don't work, check with the AMI provider.

Error using MindTerm on Safari Browser

If you use MindTerm to connect to your instance, and are using the Safari 6.1 or 7 web browser, you may get the following error:

```
Error connecting to your_instance_ip, reason:
 -> Key exchange failed: Host authentication failed
```

You need to update the browser's security settings to allow the AWS Management Console to run the Java plugin in unsafe mode.

To enable the Java plugin to run in unsafe mode

1. In Safari, keep the Amazon EC2 console open, and select **Safari**, then **Preferences**, then **Security**.
2. Click **Manage Website Settings**.
3. Select the **Java** plugin on the left, then locate the AWS Management Console URL in the **Currently Open Websites** list. Select **Run in Unsafe Mode** from its associated list.
4. When prompted, click **Trust** in the warning dialog. Click **Done** to return the browser.

Error Using Mac OS X RDP Client

If you are connecting to a Windows Server 2012 R2 instance using the Remote Desktop Connection client from the Microsoft website, you may get the following error:

```
Remote Desktop Connection cannot verify the identity of the computer that you
want to connect to.
```

Download the Microsoft Remote Desktop app from the Apple iTunes store, and use the app to connect to your instance.

Troubleshooting Stopping Your Instance

If you have stopped your Amazon EBS-backed instance and it appears "stuck" in the `stopping` state, there may be an issue with the underlying host computer.

First, try stopping the instance again. If you are using the stop-instances (AWS CLI) or ec2-stop-instances (Amazon EC2 CLI) command, be sure to use the `--force` option.

If you can't force the instance to stop, you can create an AMI from the instance and launch a replacement instance.

To create a replacement instance

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances** and select the instance.
3. Click **Actions** and then click **Create Image**.
4. In the **Create Image** dialog box, fill in the following fields and then click **Create Image**:

   a. Specify a name and description for the AMI.
   b. Select **No reboot**.

5. Launch an instance from the AMI and verify that the instance is working.

6. Select the stuck instance, click **Actions**, and then click **Terminate**. If the instance also gets stuck terminating, Amazon EC2 automatically forces it to terminate within a few hours.

If you are unable to create an AMI from the instance as described in the previous procedure, you can set up a replacement instance as follows:

To create a replacement instance (if the previous procedure fails)

1. Select the instance, open the **Description** tab, and view the **Block devices** list. Select each volume and write down its volume ID. Be sure to note which volume is the root volume.

2. In the navigation pane, click **Volumes**. Select each volume for the instance, click **Actions**, and then click **Create Snapshot**.

3. In the navigation pane, click **Snapshots**. Select the snapshot that you just created, and then select **Create Volume** from the **Actions** list.

4. Launch an instance of the same type as the stuck instance (Amazon Linux, Windows, and so on). Note the volume ID and device name of its root volume.

5. In the navigation pane, click **Instances**, select the instance that you just launched, click **Actions**, and then click **Stop**.

6. In the navigation pane, click **Volumes**, select the root volume of the stopped instance, click **Actions**, and then click **Detach Volume**.

7. Select the root volume that you created from the stuck instance, click **Attach Volume**, and attach it to the new instance as its root volume (using the device name that you wrote down). Attach any additional non-root volumes to the instance.

8. In the navigation pane, click **Instances** and select the replacement instance. Click **Actions**, and then click **Start**. Verify that the instance is working.

9. Select the stuck instance, click **Actions**, and then click **Terminate**. If the instance also gets stuck terminating, Amazon EC2 automatically forces it to terminate within a few hours.

If you're unable to complete these procedures, you can post a request for help to the Amazon EC2 forum. To help expedite a resolution, include the instance ID and describe the steps that you've already taken.

Troubleshooting Terminating (Shutting Down) Your Instance

You are not billed for any instance hours while an instance is not in the `running` state. In other words, when you terminate an instance, you stop incurring charges for that instance as soon as its state changes to `shutting-down`.

Delayed Instance Termination

If your instance remains in the `shutting-down` state longer than a few minutes, it might be delayed due to shutdown scripts being run by the instance.

Another possible cause is a problem with the underlying host computer. If your instance remains in the `shutting-down` state for several hours, Amazon EC2 treats it as a stuck instance and forcibly terminates it.

If it appears that your instance is stuck terminating and it has been longer than several hours, post a request for help to the Amazon EC2 forum. To help expedite a resolution, include the instance ID and describe the steps that you've already taken.

Automatically Launch or Terminate Instances

If you terminate all your instances, you may see that we launch a new instance for you. If you launch an instance, you may see that we terminate one of your instances. Generally, these behaviors mean that you've used Auto Scaling or AWS Elastic Beanstalk to scale your computing resources automatically based on criteria that you've defined.

For more information, see the Auto Scaling Developer Guide or the AWS Elastic Beanstalk Developer Guide.

Troubleshooting Instances with Failed Status Checks

Topics

Initial Steps You Can Take

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:

To investigate impaired instances using the AWS Management Console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**, and then select your instance.
3. Click the **Status Checks** tab in the details 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 Amazon 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 Amazon EC2 to resolve the issue.
- Post your issue to the Amazon EC2 forum.

If an instance status check fails, 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 that you 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 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.

8. If your issue is not resolved, you can post your issue to the Amazon EC2 forum.

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, verify that you 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

Device Errors

Kernel Errors

File System Errors

Operating System Errors

- Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. (SELinux misconfiguration) (p. 661)
- XENBUS: Timeout connecting to devices (Xenbus timeout) (p. 661)

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
Amazon EBS-backed	Do one of the following: • Stop the instance, and modify the instance to use a different instance type, and start the instance again. For example, a larger or a memory-optimized instance type. • Reboot the instance to return it to a non-impaired status. The problem will probably occur again unless you change the instance type.
Instance store-backed	Do one of the following: • Terminate the instance and launch a new instance, specifying a different instance type. For example, a larger or a memory-optimized instance type. • Reboot the instance to return it to an unimpaired status. The problem will probably occur again unless you change the instance type.

ERROR: mmu_update failed (Memory management 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 AWS Support.

I/O error (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
...
```

Amazon Elastic Compute Cloud User Guide for Linux
IO ERROR: neither local nor remote disk (Broken distrib-
uted block device)

Potential Causes

Instance type	Potential cause
Amazon EBS-backed	A failed Amazon EBS volume
Instance store-backed	A failed physical drive

Suggested Actions

For this instance type	Do this

Amazon EBS-backed	Use the following procedure:
	
	1. Stop the instance.
	2. Detach the volume.
	3. Attempt to recover the volume.
	
	    Tip
	    It's good practice to snapshot your Amazon EBS volumes often. This dramatically decreases the risk of data loss as a result of failure.
	
	4. Re-attach the volume to the instance.
	5. Detach the volume.

Instance store-backed	Terminate the instance and launch a new instance.
	
	    Note
	    Data cannot be recovered. Recover from backups.
	
	    Tip
	    It's a good practice to use either Amazon S3 or Amazon EBS for backups. Instance store volumes are directly tied to single host and single disk failures.

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
```

Amazon Elastic Compute Cloud User Guide for Linux
request_module: runaway loop modprobe (Looping
legacy kernel modprobe on older Linux versions)

```
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
Amazon EBS-backed	A failed Amazon EBS volume
Instance store-backed	A failed physical drive

Suggested Action

Terminate the instance and launch a new instance.

For an Amazon EBS-backed instance you can recover data from a recent snapshot by creating an image from it. Any data added after the snapshot cannot be recovered.

request_module: runaway loop modprobe (Looping legacy kernel modprobe on older Linux versions)

This condition is indicated by a system log similar to the one shown below. Using an unstable or old Linux kernel (for example, 2.6.16-xenU) 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

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c
```

Amazon Elastic Compute Cloud User Guide for Linux
"FATAL: kernel too old" and "fsck: No such file or dir-
ectory while trying to open /dev" (Kernel and AMI mis-
match)

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Use a newer kernel, either GRUB-based or static, using one of the following options. Option 1: Terminate the instance and launch a new instance, specifying the `-kernel` and `-ramdisk` parameters. Option 2: 1. Stop the instance. 2. Modify the kernel and ramdisk attributes to use a newer kernel. 3. Start the instance.
Instance store-backed	Terminate the instance and launch a new instance, specifying the `-kernel` and `-ramdisk` 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
Amazon EBS-backed	Use the following procedure: 1. Stop the instance. 2. Modify the configuration to use a newer kernel. 3. Start the instance.

Amazon Elastic Compute Cloud User Guide for Linux
"FATAL: Could not load /lib/modules" or "BusyBox"
(Missing kernel modules)

For this instance type	Do this
Instance store-backed	Use the following procedure: 1. Create an AMI that uses a newer kernel. 2. Terminate the instance. 3. Start a new instance from the AMI you created.

"FATAL: Could not load /lib/modules" or "Busy-Box" (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
- Amazon EBS root volume not correctly attached as `/dev/sda1`

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Select corrected ramdisk for the Amazon EBS volume. 2. Stop the instance. 3. Detach the volume and repair it. 4. Attach the volume to the instance. 5. Start the instance. 6. Modify the AMI to use the corrected ramdisk.
Instance store-backed	Use the following procedure: 1. Terminate the instance and launch a new instance with the correct ramdisk. 2. Create a new AMI with the correct ramdisk.

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/sda1 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
```

Amazon Elastic Compute Cloud User Guide for Linux
request_module: runaway loop modprobe (Looping
legacy kernel modprobe on older Linux versions)

```
  Booting 'Fallback'

root (hd0)

 Filesystem type is ext2fs, using whole disk

kernel /vmlinuz.old root=/dev/sda1 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
Amazon EBS-backed	Use the following procedure: 1. Stop the instance. 2. Replace with working kernel. 3. Install a fallback kernel. 4. Modify the AMI by correcting the kernel.
Instance store-backed	Use the following procedure: 1. Terminate the instance and launch a new instance with the correct kernel. 2. Create an AMI with the correct kernel. 3. (Optional) Seek technical assistance for data recovery using AWS Support.

request_module: runaway loop modprobe (Looping legacy kernel modprobe on older Linux versions)

This condition is indicated by a system log similar to the one shown below. Using an unstable or old Linux kernel (for example, 2.6.16-xenU) can cause an interminable loop condition at startup.

Amazon Elastic Compute Cloud User Guide for Linux
fsck: No such file or directory while trying to open... (File
system not found)

```
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

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c
```

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Use a newer kernel, either GRUB-based or static, using one of the following options. Option 1: Terminate the instance and launch a new instance, specifying the `-kernel` and `-ramdisk` parameters. Option 2: 1. Stop the instance. 2. Modify the kernel and ramdisk attributes to use a newer kernel. 3. Start the instance.
Instance store-backed	Terminate the instance and launch a new instance, specifying the `-kernel` and `-ramdisk` parameters.

fsck: No such file or directory while trying to open... (File system 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  ]
```

Amazon Elastic Compute Cloud User Guide for Linux
fsck: No such file or directory while trying to open... (File
system not found)

```
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
Amazon EBS-backed	Use the following procedure: 1. Stop the instance, detach the root volume, repair/modify /etc/fstab the volume, attach the volume to the instance, and start the instance. 2. Fix ramdisk to include modified /etc/fstab (if applicable). 3. Modify the AMI to use a newer ramdisk. **Tip** The sixth field in the fstab defines availability requirements of the mount – a nonzero value implies that an fsck will be done on that volume and *must* 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.
Instance store-backed	Use the following procedure: 1. Terminate the instance and launch a new instance. 2. Detach any errant Amazon EBS volumes and the reboot instance. 3. (Optional) Seek technical assistance for data recovery using AWS Support.

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
Amazon EBS-backed	• Detached or failed Amazon EBS volume. • Corrupted filesystem. • Mismatched ramdisk and AMI combination (e.g., Debian ramdisk with a SUSE AMI).
Instance store-backed	• A failed drive. • A corrupted file system. • A mismatched ramdisk and combination (for example, a Debian ramdisk with a SUSE AMI).

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Stop the instance. 2. Detach the root volume. 3. Attach the root volume to a known working instance. 4. Run filesystem check (fsck –a /dev/...). 5. Fix any errors. 6. Detach the volume from the known working instance. 7. Attach the volume to the stopped instance. 8. Start the instance. 9. Recheck the instance status.
Instance store-backed	Try one of the following: • Start a new instance. • (Optional) Seek technical assistance for data recovery using AWS Support.

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/sda1 ro 4
...
Registering block device major 8
...
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(8,1)
```

Amazon Elastic Compute Cloud User Guide for Linux
Error: Unable to determine major/minor number of root
device... (Root file system/device mismatch)

Potential Causes

Instance type	Potential cause
Amazon EBS-backed	• Device not attached correctly. • Root device not attached at correct device point. • Filesystem not in expected format. • Use of legacy kernel (e.g., 2.6.16-XenU).
Instance store-backed	Hardware device failure.

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Do one of the following: • Stop and then restart the instance. • Modify root volume to attach at the correct device point, possible /dev/sda1 instead of /dev/sda. • Stop and modify to use modern kernel.
Instance store-backed	Terminate the instance and launch a new instance using a modern kernel.

Error: Unable to determine major/minor number of root device... (Root file system/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
Amazon EBS-backed	Use the following procedure: 1. Stop the instance. 2. Detach the volume. 3. Fix the device mapping problem. 4. Start the instance. 5. Modify the AMI to address device mapping issues.
Instance store-backed	Use the following procedure: 1. Create a new AMI with the appropriate fix (map block device correctly). 2. Terminate the instance and launch a new instance from the AMI you created.

XENBUS: Device with no driver...

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)
Initalizing 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
Amazon EBS-backed	Use the following procedure: 1. Stop the instance. 2. Detach the volume. 3. Fix the device mapping problem. 4. Start the instance. 5. Modify the AMI to address device mapping issues.
Instance store-backed	Use the following procedure: 1. Create an AMI with the appropriate fix (map block device correctly). 2. Terminate the instance and launch a new instance using the AMI you created.

... days without being checked, check forced (File system check required)

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
Amazon EBS-backed	Try one or more of the following to resolve the issue: • Stop the instance, attach the volume to an existing running instance. • Manually run consistency checks. • Fix ramdisk to include relevant utilities. • Modify filesystem tuning parameters to remove consistency requirements (not recommended).

For this instance type	Do this
Instance store-backed	Try one or more of the following to resolve the issue: • Rebundle ramdisk with correct tooling. • Modify file system tuning parameters to remove consistency requirements (not recommended). • Terminate the instance and launch a new instance. • (Optional) Seek technical assistance for data recovery using AWS Support.

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
Amazon EBS-backed	• Missing GRUB configuration file. • Incorrect GRUB image used, expecting GRUB configuration file at a different location. • Unsupported filesystem used to store your GRUB configuration file.
Instance store-backed	• Missing GRUB configuration file. • Incorrect GRUB image used, expecting GRUB configuration file at a different location.

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Option 1: Modify the AMI and relaunch the instance 1. Modify the source AMI to create a GRUB configuration file at the standard location (/boot/grub/menu.lst). 2. Pick the appropriate GRUB image, (hd0-1st drive or hd00 – 1st drive, 1st partition). 3. Terminate the instance and launch a new one using the AMI the you created. Option 2: Fix the existing instance 1. Stop the instance. 2. Detach the root filesystem. 3. Attach the root filesystem to a known working instance. 4. Mount filesystem. 5. Create a GRUB configuration file. 6. Detach filesystem. 7. Attach to the original instance. 8. Modify kernel attribute to use the appropriate GRUB image (1st disk or 1st partition on 1st disk). 9. Start the instance.
Instance store-backed	Option 1: Modify the AMI and relaunch the instance: 1. Create the new AMI with a GRUB configuration file at the standard location (/boot/grub/menu.lst). 2. Pick the appropriate GRUB image, (hd0-1st drive or hd00 – 1st drive, 1st partition). 3. Terminate the instance and launch a new instance using the AMI you created. Option 2: Terminate the instance and launch a new instance, specifying the correct kernel. **Note** To recover data from the existing instance, contact AWS Support.

Amazon Elastic Compute Cloud User Guide for Linux
Bringing up interface eth0: Device eth0 has different
MAC address than expected, ignoring. (Hard-coded MAC
address)

Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (Hard-coded 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 hard-coded interface MAC in the AMI configuration.

Suggested Actions

For this instance type	Do this
Amazon EBS-backed	Do one of the following: • Modify the AMI to remove the hard coding and relaunch the instance. • Modify the instance to remove the hard-coded MAC address. OR Use the following procedure: 1. Stop the instance. 2. Detach the root volume. 3. Attach the volume to another instance and modify the volume to remove the hard-coded MAC address. 4. Attach the volume to the original instance. 5. Start the instance.
Instance store-backed	Do one of the following: • Modify the instance to remove the hard-coded MAC address. • Terminate the instance and launch a new instance.

Amazon Elastic Compute Cloud User Guide for Linux
Unable to load SELinux Policy. Machine is in enforcing
mode. Halting now. (SELinux misconfiguration)

Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. (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
Amazon EBS-backed	Use the following procedure: 1. Stop the instance. 2. Detach the volume. 3. Disable SELinux. 4. Start the instance.
Instance store-backed	Use the following procedure: 1. Terminate the instance and launch a new instance. 2. (Optional) Seek technical assistance for data recovery using AWS Support.

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
Amazon EBS-backed	Do one of the following: • Modify the AMI and instance to use a modern kernel and relaunch the instance. • Reboot the instance.
Instance store-backed	Do one of the following: • Terminate the instance. • Modify the AMI to use a modern kernel, and launch a new instance using this AMI.

Troubleshooting Instance Capacity

The following errors are related to instance capacity.

Error: InsufficientInstanceCapacity

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.

Error: InstanceLimitExceeded

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 default limit is 20. If you need additional running instances, complete the form at Request to Increase Amazon EC2 Instance Limit.

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 support these tasks through the Amazon EC2 API and the command line interface tools (CLI).

Instance Reboot

Just as you can reset a computer by pressing the reset button, you can reset Amazon EC2 instances using the Amazon EC2 console, CLI, or API. For more information, see Reboot Your Instance (p. 312).

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

Instance Console Output

For Linux/Unix instances, the instance console output displays the exact console output that would normally be displayed on a physical monitor attached to a computer. 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 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.

Only the instance owner can access the console output. You can retrieve the console output for your instances using the console or the command line.

To get console output using the console

1. Open the Amazon EC2 console.
2. Select the instance.
3. Click **Actions** and then click **Get System Log**.

To get console output using the command line

You can use one of the following commands. For more information about these command line interfaces, see Accessing Amazon EC2 (p. 3).

- get-console-output (AWS CLI)
- ec2-get-console-output (Amazon EC2 CLI)

Instance Recovery When its Host Computer Fails

If there is an unrecoverable issue with the hardware of an underlying host computer, AWS may schedule an instance stop event. You'll be notified of such an event ahead of time by email.

If you have an Amazon EBS-backed instance running on a host computer that fails, you can do the following to recover:

1. Back up any important data on your instance store volumes to Amazon EBS or Amazon S3.
2. Stop the instance.
3. Start the instance.
4. Restore any important data.
5. [EC2-Classic] If the instance had an associated Elastic IP address, you must reassociate it with the instance.

For more information, see Stop and Start Your Instance (p. 309).

If you have an instance store-backed instance running on a host computer that fails, you can do the following to recover:

1. Create an AMI from the instance.
2. Upload the image to Amazon S3.
3. Back up important data to Amazon EBS or Amazon S3.
4. Terminate the instance.
5. Launch a new instance from the AMI.
6. Restore any important data to the new instance.
7. [EC2-Classic] If the original instance had an associated Elastic IP address, you must associate it with the new instance.

For more information, see Creating an Instance Store-Backed Linux AMI (p. 77).

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.

To monitor the calls made to the Amazon EC2 API for your account, including calls made by the AWS Management Console, command line tools, and other services, use AWS CloudTrail. For more information, see the AWS CloudTrail User Guide.

Topics

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++.

Available APIs for Amazon EC2

The Amazon EC2 Query API provides HTTP or HTTPS requests that use the HTTP verb GET or POST and a Query parameter named `Action`. For more information about these API actions, see the Amazon EC2 API Reference.

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 about downloading the AWS SDKs, see AWS SDKs and Tools. For more information about the language-specific APIs for Amazon EC2, see the following documentation.

AWS SDK for .NET

- Amazon.EC2
- Amazon.EC2.Model
- Amazon.EC2.Util

AWS SDK for Java

- com.amazonaws.services.ec2
- com.amazonaws.services.ec2.model
- com.amazonaws.services.ec2.util

AWS SDK for JavaScript

- AWS.EC2

AWS SDK for Python

- boto.ec2

AWS SDK for Ruby

- AWS::EC2

AWS SDK for PHP

- Ec2Client

AWS SDK for IOS

- AWSEC2

AWS SDK for Android

- com.amazonaws.services.ec2
- com.amazonaws.services.ec2.model
- com.amazonaws.services.ec2.util

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

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 request 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 `RunInstance` to run an instance).
- **Parameters**: Any parameters for the action; each parameter is separated by an ampersand (&). You must also include authorization parameters that AWS uses to ensure the validity and authenticity of the request. Amazon EC2 supports Signature Version 2 and Signature Version 4. For more information, see Query API Authentication (p. 668).

The following is an example request that launches instances:

```
https://ec2.amazonaws.com/?Action=RunInstances&ImageId=ami-2bb65342&Max
Count=3&MinCount=1&Placement.AvailabilityZone=us-east-1a&Monitoring.En
abled=true&Version=2014-10-01&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Creden
tial=AKIDEXAMPLE%2F20130813%2Fus-east-1%2Fec2%2Faws4_request&X-Amz-
Date=20130813T150206Z&X-Amz-SignedHeaders=content-type%3host%3x-amz-date&X-Amz-
Signature=525d1a96c69b5549dd78dbbec8efe264102288b83ba87b7d58d4b76b71f59fd2
Content-type: application/json
host:ec2.amazonaws.com
```

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-1a
&Monitoring.Enabled=true
&Version=2014-10-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLEus-east-1%2Fec2%2Faws4_request
&X-Amz-Date=20130813T150206Z
&X-Amz-SignedHeaders=content-type%3host%3x-amz-date
&X-Amz-Signature=ced6826de92d2bdeed8f846f0bf508e8559e98e4b0194b84ex
ample54174deb456c
```

```
Content-type: application/json
host:ec2.amazonaws.com
```

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 EC2 API Reference*.

The remaining lines specify additional parameters for the request.

> **Important**
> Before you specify your access key ID for the `AWSAccessKeyId` or `Credential` parameter, review and follow the guidance in Best Practices for Managing AWS Access Keys.

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. 7). 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.

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 EC2 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 the HTTP or HTTPS protocol.

Regardless of which protocol you use, you must include a signature in every Query request. Amazon EC2 supports Signature Version 2 and Signature Version 4. For more information, see Signature Version

2 Signing Process and Signature Version 4 Signing Process in the *Amazon Web Services General Reference*.

Signature Version 4 requests allow you specify all the authorization parameters in a single header, for example:

```
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Amz-Date: 20130813T150211Z
Host: ec2.amazonaws.com
Authorization: AWS4-HMAC-SHA256 Credential=AKIDEXAMPLE/20130813/us-east-
1/ec2/aws4_request, SignedHeaders=content-type;host;x-amz-date, Signa
ture=ced6826de92d2bdeed8f846f0bf508e8559e98e4b0194b84example54174deb456c

http://ec2.amazonaws.com/?Action=RunInstances
ImageId=ami-2bb65342
&MaxCount=3
&MinCount=1
&Monitoring.Enabled=true
&Placement.AvailabilityZone=us-east-1a
&Version=2014-10-01
```

In the example Query requests we present in the Amazon EC2 documentation, we omit headers and 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.

```
<DescribeKeyPairsResponse xmlns="http://ec2.amazonaws.com/doc/2014-10-01/">
  <requestId>7a62c49f-347e-4fc4-9331-6e8eEXAMPLE</requestId>
  <keySet>
    <item>
      <keyName>gsg-keypair</keyName>
      <keyFingerprint>
         00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
      </keyFingerprint>
    </item>
  </keySet>
</DescribeKeyPairsResponse>
```

To download the WSDL file for Amazon EC2, see Amazon EC2 Developer Resources.

Troubleshooting API Request Errors

In the Amazon EC2 Query API, errors codes are indicated as being either client or server. Client errors usually occur because there is a problem with the structure, content, or validity of the request. Server errors usually indicate a server-side issue.

For more information about API error codes, see Error Codes in the *Amazon EC2 API Reference*.

Topics

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.

Calculating the sleep interval

When you have to poll or retry an API request, we recommend using an exponential backoff algorithm to calculate the sleep interval between API calls. The idea behind exponential backoff is to use progressively longer waits between retries for consecutive error responses. For more information, and implementation examples of this algorithm, see Error Retries and Exponential Backoff in AWS.

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 all 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. Run the appropriate `Describe` command using an exponential backoff 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 five minutes of wait time.
- Add wait time between subsequent commands, even if a `Describe` command returns an accurate response. Apply an exponential backoff algorithm starting with a couple of seconds of wait time, and increase gradually up to about five minutes of wait time.

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 run the `RunInstances` command, and then immediately run another command using the instance ID that was provided in the response of `RunInstances`, it may return an `Invalid-InstanceID.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 run the `CreateSecurityGroup` command, and then immediately run another command 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.

Unauthorized Operation

By default, AWS Identity and Access Management (IAM) users don't have permission to create or modify Amazon EC2 resources, or perform tasks using the Amazon EC2 API, unless they've been explicitly

granted permission through IAM policies. If an IAM user attempts to perform an action for which permission has not been granted, the request returns the following error: `Client.UnauthorizedOperation`.

This error may occur when a policy is unintentionally restrictive. For example, to allow an IAM user to launch instances into a specific subnet, you need to grant permissions for the following resources by specifying their ARNs in your IAM policy: instances, volumes, AMIs, the specific subnet, network interfaces, key pairs, and security groups. If you omit the permission for volumes, for example, the user is only able to launch an instance from an instance store-backed AMI, as they do not have permission to create the root EBS volume for an EBS-backed instance.

For more information about creating IAM policies for Amazon EC2, see IAM Policies for Amazon EC2 (p. 431).

Currently, not all API actions support resource-level permissions; we'll add support for more in 2014. For more information about which ARNs you can use with which Amazon EC2 API actions, see Granting IAM Users Required Permissions for Amazon EC2 Resources in the *Amazon EC2 API Reference*.

Ensuring Idempotency

An *idempotent* operation completes no more than one time.

When you launch an instance, the request typically returns before the operation has completed. You determine whether the operation was successful by monitoring the state of the instance (it goes from `pending` to `running`). If the operation times out or there are connection issues, you might need to retry the request. However, if the original request and a retry are both successful, you'll end up with more instances than you intended to launch.

If you launch your instance using run-instances (AWS CLI), ec2-run-instances (Amazon EC2 CLI), or RunInstances, you can optionally provide a client token to ensure that the request is idempotent. If you repeat a request, the same response is returned for each repeated request. The only information that might vary in the response is the state of the instance.

The client token is a unique, case-sensitive string of up to 64 ASCII characters. It is included in the response when you describe the instance. The client token is valid for at least 24 hours after the termination of the instance. You should not reuse a client token in another call later on.

If you repeat a request with the same client token, but change another request parameter, Amazon EC2 returns an `IdempotentParameterMismatch` error.

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 then use the same client token in a request in other regions, we'll launch instances in each of those regions.

The following table shows common response codes and the recommended course of action.

Code	Retry	Comments
200 (OK)	No effect	The request has succeeded and any further retries have no effect.
400 (Client Error)	Not recommended	The request will never succeed (for example, a specified parameter value is not valid). If the request involves a resource that is in the process of changing states, repeating the request could possibly succeed (for example, launching an instance using an Amazon EBS volume that is about to become `available`).

Code	Retry	Comments
500 (Server Internal Error)	Recommended	The error is generally transient. Repeat the request with an appropriate back-off strategy.
503 (Server Unavailable)	Recommended	The error can occur when there is extreme load. Repeat the request with an appropriate back-off strategy.

Idempotency Support

The following commands and actions are idempotent:

- associate-address (AWS CLI)
- AssociateAddress
- disassociate-address (AWS CLI)
- DisassociateAddress
- ec2-associate-address (Amazon EC2 CLI)
- ec2-disassociate-address (Amazon EC2 CLI)
- ec2-terminate-instances (Amazon EC2 CLI)
- terminate-instances (AWS CLI)
- TerminateInstances

The following commands and actions support idempotent operations using a client token:

- copy-image (AWS CLI)
- CopyImage
- create-reserved-instances-listing (AWS CLI)
- CreateReservedInstancesListing
- ec2-copy-image (Amazon EC2 CLI)
- ec2-create-reserved-instances-listing (Amazon EC2 CLI)
- ec2-modify-reserved-instances (Amazon EC2 CLI)
- ec2-run-instances (Amazon EC2 CLI)
- modify-reserved-instances (AWS CLI)
- ModifyReservedInstances
- run-instances (AWS CLI)
- RunInstances

Example Idempotent Command

To make a command an idempotent request, add the `--client-token` option. The client token is a unique, case-sensitive string of up to 64 ASCII characters.

AWS CLI

Use the run-instances command as follows to make an idempotent request:

```
aws ec2 run-instances --image-id ami-b232d0db --count 1 --key-name my-key-pair
 --client-token 550e8400-e29b-41d4-a716-446655440000
```

Amazon EC2 CLI

Use the ec2-run-instances command as follows to make an idempotent request:

```
ec2-run-instances ami-b232d0db -k my-key-pair --client-token 550e8400-e29b-41d4-
a716-446655440000
```

Example Idempotent Query

Use the RunInstances action as follows to make an idempotent request:

```
https://ec2.amazonaws.com/?Action=RunInstances
&ImageId=ami-3ac33653
&MaxCount=1
&MinCount=1
&KeyName=my-key-pair
&ClientToken=550e8400-e29b-41d4-a716-446655440000
&AUTHPARAMS
```

The *ClientToken* parameter requires a unique, case-sensitive string of up to 64 ASCII characters.

SOAP Requests

We have deprecated the SOAP API for Amazon EC2. We will continue to support SOAP requests for API versions up to and including version 2014-02-01, until the end of December 2014. If you use a SOAP request against a later API version, or after December 2014, you will receive the following response:

```
Client.UnsupportedProtocol: SOAP is no longer supported.
```

Similarly, the AWS software development kits (SDKs) will continue to support SOAP requests for Amazon EC2 API versions up to and including version 2014-02-01, until the end of December 2014.

If you are using the Amazon EC2 CLI tools, you will be able to use the EC2_PRIVATE_KEY and EC2_CERT environment variables for versions up to and including version 1.6.13.0. Thereafter, you must use the AWS_ACCESS_KEY and AWS_SECRET_KEY variables instead. For more information, see Setting Up the Amazon EC2 CLI and AMI Tools.

We recommend that you use the Query API for Amazon EC2, or the SDKs for AWS. For more information, see Making API Requests (p. 665).

Logging Amazon EC2 API Calls Using AWS CloudTrail

Amazon EC2 and Amazon VPC are integrated with CloudTrail, a service that captures API calls made by or on behalf of Amazon EC2 and Amazon VPC and delivers the log files to an Amazon S3 bucket that you specify. The API calls can be made indirectly by using the Amazon EC2 or Amazon VPC console,

or directly by using the Amazon EC2 API. Using the information collected by CloudTrail, you can determine what request was made, the source IP address from which the request was made, who made the request, when it was made, and so on. To learn more about CloudTrail, including how to configure and enable it, see the AWS CloudTrail User Guide.

Amazon EC2 Information in CloudTrail

When CloudTrail logging is enabled, calls made to Amazon EC2 and Amazon VPC actions are tracked in log files, along with any other AWS service records. CloudTrail determines when to create and write to a new file based on a specified time period and file size.

All of the Amazon EC2 and Amazon VPC actions are logged, and are documented in the Amazon EC2 API Reference. For example, calls to the RunInstances, DescribeInstances, or CreateImage API actions generate entries in the CloudTrail log files.

Every log entry contains information about who generated the request. The user identity information in the log helps you determine whether the request was made with root or IAM user credentials, with temporary security credentials for a role or federated user, or by another AWS service. For more information, see the **userIdentity** field in the CloudTrail Event Reference.

You can store your log files in your bucket for as long as you want, but you can also define Amazon S3 lifecycle rules to archive or delete log files automatically. By default, your log files are encrypted by using Amazon S3 server-side encryption (SSE).

You can choose to have CloudTrail publish Amazon SNS notifications when new log files are delivered if you want to take quick action upon log file delivery. For more information, see Configuring Amazon SNS Notifications.

You can also aggregate Amazon EC2 and Amazon VPC log files from multiple AWS regions and multiple AWS accounts into a single Amazon S3 bucket. For more information, see Aggregating CloudTrail Log Files to a Single Amazon S3 Bucket.

Understanding Amazon EC2 Log File Entries

CloudTrail log files can contain one or more log entries where each entry is made up of multiple JSON-formatted events. A log entry represents a single request from any source and includes information about the requested action, any input parameters, the date and time of the action, and so on. The log entries are not in any particular order. That is, they are not an ordered stack trace of the public API calls.

The following log file record shows that a user terminated two instances.

```
{
    "Records":[
        {
            "eventVersion":"1.01",
            "userIdentity":{
                "type":"Root",
                "principalId":"111122223333",
                "arn":"arn:aws:iam::111122223333:user/testuser",
                "accountId":"111122223333",
                "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
                "sessionContext":{
                    "attributes":{
                        "mfaAuthenticated":"false",
                        "creationDate":"2014-05-02T08:27:22Z"
                    }
                }
```

```
            },
            "eventTime":"2014-05-02T08:27:45Z",
            "eventSource":"ec2.amazonaws.com",
            "eventName":"TerminateInstances",
            "awsRegion":"us-east-1",
            "sourceIPAddress":"192.0.2.64",
            "userAgent":"EC2ConsoleBackend, aws-sdk-java/unknown-version
Linux/2.6.18-308.24.1.1123.6.fleetxen Java_HotSpot(TM)_64-Bit_Server_VM/24.51-
b03",
            "requestParameters":{
                "instancesSet":{
                    "items":[
                        {
                            "instanceId":"i-1a2b3c4d"
                        },
                        {
                            "instanceId":"i-111bbb33"
                        }
                    ]
                }
            },
            "responseElements":{
                "instancesSet":{
                    "items":[
                        {
                            "instanceId":"i-1a2b3c4d",
                            "currentState":{
                                "code":48,
                                "name":"terminated"
                            },
                            "previousState":{
                                "code":48,
                                "name":"terminated"
                            }
                        },
                        {
                            "instanceId":"i-111bbb33",
                            "currentState":{
                                "code":48,
                                "name":"terminated"
                            },
                            "previousState":{
                                "code":48,
                                "name":"terminated"
                            }
                        }
                    ]
                }
            },
            "requestID":"be112233-1ba5-4ae0-8e2b-1c302example",
            "eventID":"6e12345-2a4e-417c-aa78-7594fexample"
        }
    ]
}
```

Document History

The following table describes important additions to the Amazon EC2 documentation. We also update the documentation frequently to address the feedback that you send us.

Current API version: 2014-10-01.

Feature	API Version	Description	Release Date
`DescribeVolumes` pagination support	2014-09-01	The `DescribeVolumes` API call now supports the pagination of results with the `MaxResults` and `NextToken` parameters. For more information, see DescribeVolumes in the *Amazon EC2 API Reference*.	23 October 2014
T2 Instances	2014-06-15	T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see T2 Instances (p. 105).	30 June 2014
New **EC2 Service Limits** page		Use the **EC2 Service Limits** page in the Amazon EC2 console to view the current limits for resources provided by Amazon EC2 and Amazon VPC, on a per-region basis.	19 June 2014
Amazon EBS General Purpose (SSD) Volumes	2014-05-01	General Purpose (SSD) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies, the ability to burst to 3,000 IOPS for extended periods of time, and a base performance of 3 IOPS/GiB. General Purpose (SSD) volumes can range in size from 1 GiB to 1 TiB. For more information, see General Purpose (SSD) Volumes (p. 524).	16 June 2014

Feature	API Version	Description	Release Date
Amazon EBS encryption	2014-05-01	Amazon EBS encryption offers seamless encryption of EBS data volumes and snapshots, eliminating the need to build and maintain a secure key management infrastructure. EBS encryption enables data at rest security by encrypting your data using Amazon-managed keys. The encryption occurs on the servers that host EC2 instances, providing encryption of data as it moves between EC2 instances and EBS storage. For more information, see Amazon EBS Encryption (p. 564).	21 May 2014
R3 Instances	2014-02-01	Next generation memory-optimized instances with the best price point per GiB of RAM and high performance. These instances are ideally suited for relational and NoSQL databases, in-memory analytics solutions, scientific computing, and other memory-intensive applications that can benefit from the high memory per vCPU, high compute performance, and enhanced networking capabilities of R3 instances. For more information about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details.	9 April 2014
New Amazon Linux AMI release		Amazon Linux AMI 2014.03 is released.	27 March 2014
Amazon EC2 Usage Reports		Amazon EC2 Usage Reports is a set of reports that shows cost and usage data of your usage of EC2. For more information, see Amazon EC2 Usage Reports (p. 619).	28 January 2014
Additional M3 instances	2013-10-15	The M3 instance sizes `m3.medium` and `m3.large` are now supported. For more information about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details.	20 January 2014
I2 instances	2013-10-15	These instances provide very high IOPS and support TRIM on Linux instances for better successive SSD write performance. I2 instances also support enhanced networking that delivers improve inter-instance latencies, lower network jitter, and significantly higher packet per second (PPS) performance. For more information, see I2 Instances (p. 109).	19 December 2013
Updated M3 instances	2013-10-15	The M3 instance sizes, `m3.xlarge` and `m3.2xlarge` now support instance store with SSD volumes. For more information about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details.	19 December 2013

Feature	API Version	Description	Release Date
Importing Linux virtual machines	2013-10-15	The VM Import process now supports the importation of Linux instances. For more information, see VM Import/Export Prerequisites (p. 268).	16 December 2013
Resource-level permissions for RunInstances	2013-10-15	You can now create policies in AWS Identity and Access Management to control resource-level permissions for the Amazon EC2 RunInstances API action. For more information and example policies, see Controlling Access to Amazon EC2 Resources (p. 429).	20 November 2013
C3 instances	2013-10-15	Compute-optimized instances that provide very high CPU performance at an economical price. C3 instances also support enhanced networking that delivers improved inter-instance latencies, lower network jitter, and significantly higher packet per second (PPS) performance. These instances are ideally suited for high-traffic web applications, ad serving, batch processing, video encoding, distributed analytics, high-energy physics, genome analysis, and computational fluid dynamics. For more information about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details.	14 November 2013
Launching an instance from the AWS Marketplace		You can now launch an instance from the AWS Marketplace using the Amazon EC2 launch wizard. For more information, see Launching an AWS Marketplace Instance (p. 297).	11 November 2013
G2 instances	2013-10-01	These instances are ideally suited for video creation services, 3D visualizations, streaming graphics-intensive applications, and other server-side workloads requiring massive parallel processing power. For more information, see Linux GPU Instances (p. 115).	4 November 2013
New launch wizard		There is a new and redesigned EC2 launch wizard. For more information, see Launching an Instance (p. 290).	10 October 2013
Modifying Instance Types of Amazon EC2 Reserved Instances	2013-10-01	You can now modify the instance type of Linux Reserved Instances within the same family (for example, M1, M2, M3, C1). For more information, see Modifying Your Reserved Instances (p. 218).	09 October 2013
New Amazon Linux AMI release		Amazon Linux AMI 2013.09 is released.	30 September 2013
Modifying Amazon EC2 Reserved Instances	2013-08-15	You can now modify Reserved Instances in a region. For more information, see Modifying Your Reserved Instances (p. 218).	11 September 2013

Feature	API Version	Description	Release Date
Assigning a public IP address	2013-07-15	You can now assign a public IP address when you launch an instance in a VPC. For more information, see Assigning a Public IP Address (p. 484).	20 August 2013
Granting resource-level permissions	2013-06-15	Amazon EC2 supports new Amazon Resource Names (ARNs) and condition keys. For more information, see IAM Policies for Amazon EC2 (p. 431).	8 July 2013
Incremental Snapshot Copies	2013-02-01	You can now perform incremental snapshot copies. For more information, see Copying an Amazon EBS Snapshot (p. 561).	11 June 2013
New **Tags** page		There is a new **Tags** page in the Amazon EC2 console. For more information, see Tagging Your Amazon EC2 Resources (p. 610).	04 April 2013
New Amazon Linux AMI release		Amazon Linux AMI 2013.03 is released.	27 March 2013
Additional EBS-optimized instance types	2013-02-01	The following instance types can now be launched as EBS-optimized instances: `c1.xlarge`, `m2.2xlarge`, `m3.xlarge`, and `m3.2xlarge`. For more information, see Amazon EBS–Optimized Instances (p. 125).	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 Copying an AMI (p. 85).	11 March 2013
Launch instances into a default VPC	2013-02-01	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. For more information, see Supported Platforms (p. 471).	11 March 2013
High-memory cluster (cr1.8xlarge) instance type	2012-12-01	Have large amounts of memory coupled with high CPU and network performance. These instances are well suited for in-memory analytics, graph analysis, and scientific computing applications.	21 January 2013
High storage (`hs1.8xlarge`) instance type	2012-12-01	High storage instances provide very high storage density and high sequential read and write performance per instance. They are well-suited for data warehousing, Hadoop/MapReduce, and parallel file systems. For more information, see HS1 Instances (p. 112).	20 December 2012

Feature	API Version	Description	Release Date
EBS snapshot copy	2012-12-01	You can use snapshot copies to create backups of data, to create new Amazon EBS volumes, or to create Amazon Machine Images (AMIs). For more information, see Copying an Amazon EBS Snapshot (p. 561).	17 December 2012
Updated EBS metrics and status checks for Provisioned IOPS (SSD) volumes	2012-10-01	Updated the EBS metrics to include two new metrics for Provisioned IOPS (SSD) volumes. For more information, see Monitoring Volumes with CloudWatch (p. 534). Also added new status checks for Provisioned IOPS (SSD) volumes. For more information, see Monitoring Volumes with Status Checks (p. 537).	20 November 2012
Linux Kernels		Updated AKI IDs; reorganized distribution kernels; updated PVOps section.	13 November 2012
M3 Instances	2012-10-01	There are new M3 extra-large and M3 double-extra-large instance types. For more information about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details.	31 October 2012
Amazon EC2 Spot Instance Request Status	2012-10-01	Spot Instance request status makes it easy to determine the state of your Amazon EC2 Spot requests.	14 October 2012
New Amazon Linux AMI release		Amazon Linux AMI 2012.09 is released.	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) (SSD) for Amazon EBS	2012-07-20	Provisioned IOPS (SSD) volumes deliver predictable, high performance for I/O intensive workloads, such as database applications, that rely on consistent and fast response times. For more information, see Amazon EBS Volume Types (p. 523).	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 HI1 Instances (p. 110).	18 July 2012

Feature	API Version	Description	Release Date
IAM roles on Amazon EC2 instances	2012-06-01	IAM roles for Amazon EC2 provide: • AWS access keys for applications running on Amazon EC2 instances. • Automatic rotation of the AWS access keys on the Amazon EC2 instance. • Granular permissions for applications running on Amazon EC2 instances that make requests to your AWS services.	11 June 2012
Spot Instance features that make it easier to get started and handle the potential of interruption.		Using Auto Scaling, you can now manage your Spot Instances: • 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 Managing Spot Instances with Auto Scaling (p. 161). • Get notifications when instances are launched or terminated. For more information, see Using Auto Scaling to Get Notifications for Spot Instances (p. 173). • Use AWS CloudFormation templates to launch Spot Instances in a stack with AWS resources. For more information, see Using CloudFormation Templates to Launch Spot Instances (p. 176).	7 June 2012
EC2 instance export and timestamps for status checks for Amazon EC2	2012-05-01	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 a VPC.	26 April 2012
AWS Marketplace AMIs	2012-04-01	Added support for AWS Marketplace AMIs.	19 April 2012
New Linux AMI release		Amazon Linux AMI 2012.03 is released.	28 March 2012
New AKI version		We've released AKI version 1.03 and AKIs for the AWS GovCloud (US) region.	28 March 2012

Feature	API Version	Description	Release Date
Medium instances, support for 64-bit on all AMIs, and a Java-based SSH Client	2011-12-15	Added support for a new instance type and 64-bit information. Added procedures for using the Java-based SSH client to connect to Linux instances.	7 March 2012
Reserved Instance pricing tiers	2011-12-15	Added a new section discussing how to take advantage of the discount pricing that is built into the Reserved Instance pricing tiers. For more information, see Understanding Reserved Instance Pricing Tiers (p. 193).	5 March 2012
Elastic Network Interfaces (ENIs) for EC2 instances in Amazon Virtual Private Cloud	2011-12-01	Added new section about elastic network interfaces (ENIs) for EC2 instances in a VPC. For more information, see Elastic Network Interfaces (ENI) (p. 494).	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: *Heavy Utilization*, *Medium Utilization*, and *Light Utilization*. For more information, see Reserved Instances (p. 185).	01 December 2011
Amazon EC2 instance status	2011-11-01	You can view additional details about the status of your instances, including scheduled events planned by AWS that might have an impact on your instances. These operational activities include instance reboots required to apply software updates or security patches, or instance retirements required where there are hardware issues. For more information, see Monitoring the Status of Your Instances (p. 346).	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 AKIs for the new US-West 2 Region.	8 November 2011
Amazon EC2 Spot Instances in Amazon VPC	2011-07-15	Added information about the support for Amazon EC2 Spot Instances in Amazon VPC. With this update, users will be able to launch Spot Instances in the Amazon Virtual Private Cloud (Amazon VPC). By launching Spot Instances in Amazon VPC, users of Spot Instances can enjoy all of the controls and advanced security options of Amazon VPC. For more information, see Launching Spot Instances in Amazon Virtual Private Cloud (p. 177).	11 October 2011

Feature	API Version	Description	Release Date
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.	26 September 2011
Simplified VM import process for users of the CLI tools	2011-07-15	The VM Import process for CLI users is simplified with the enhanced functionality of `ec2-import-instance` and `ec2-import-volume`, which now will perform the upload of the images into Amazon EC2 after creating the import task. In addition, with the introduction of the `ec2-resume-import` command, users can restart an incomplete upload at the point the task stopped. For more information, see Step 4: Importing Your VM into Amazon EC2 (p. 273).	15 September 2011
Support for importing in VHD file format		VM Import can now import virtual machine image files in VHD format. The VHD file format is compatible with the Citrix Xen and Microsoft Hyper-V virtualization platforms. With this release, VM Import now supports RAW, VHD and VMDK (VMware ESX-compatible) image formats. For more information, see Step 1: Install the Amazon EC2 CLI (p. 271).	24 August 2011
Update to the Amazon EC2 VM Import Connector for VMware vCenter		Added information about the 1.1 version of the Amazon EC2 VM Import Connector for VMware vCenter virtual appliance (Connector). This update includes proxy support for Internet access, better error handling, improved task progress bar accuracy, and several bug fixes. For more information, see Importing a VM into Amazon EC2 (p. 270).	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, see PV-GRUB (p. 98).	20 June 2011
Spot Instances Availability Zone pricing changes	2011-05-15	Added information about the Spot Instances Availability Zone pricing feature. In this release, we've added new Availability Zone pricing options as part of the information returned when you query for Spot Instance requests and Spot Price history. These additions make it easier to determine the price required to launch a Spot Instance into a particular Availability Zone. For more information, see Spot Instances (p. 132).	26 May 2011

Feature	API Version	Description	Release Date
AWS Identity and Access Management		Added information about AWS Identity and Access Management (IAM), which enables users to specify which Amazon EC2 actions a user can use with Amazon EC2 resources in general. For more information, see Controlling Access to Amazon EC2 Resources (p. 429).	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, see PV-GRUB (p. 98).	26 April 2011
Dedicated instances		Launched within your Amazon Virtual Private Cloud (Amazon VPC), Dedicated Instances are instances that are physically isolated at the host hardware level. Dedicated Instances let you take advantage of Amazon VPC and the AWS cloud, with benefits including on-demand elastic provisioning and pay only for what you use, while isolating your Amazon EC2 compute instances at the hardware level. For more information, see Using EC2 Dedicated Instances in the *Amazon VPC User Guide*.	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 Reserved Instances (p. 185).	27 March 2011
New Amazon Linux reference AMI		The new Amazon Linux reference AMI 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 AMIs for GPU Instances (p. 116).	15 March 2011
Metadata information	2011-01-01	Added information about metadata to reflect changes in the 2011-01-01 release. For more information, see Instance Metadata and User Data (p. 256) and Instance Metadata Categories (p. 262).	11 March 2011
Amazon EC2 VM Import Connector for VMware vCenter		Added information about the Amazon EC2 VM Import Connector for VMware vCenter virtual appliance (Connector). The Connector is a plug-in for VMware vCenter that integrates with VMware vSphere Client and provides a graphical user interface that you can use to import your VMware virtual machines to Amazon EC2. For more information, see Importing a VM into Amazon EC2 (p. 270).	3 March 2011

Feature	API Version	Description	Release Date
Force volume detachment		You can now use the AWS Management Console to force the detachment of an Amazon EBS volume from an instance. For more information, see Detaching an Amazon EBS Volume from an Instance (p. 544).	23 February 2011
Instance termination protection		You can now use the AWS Management Console to prevent an instance from being terminated. For more information, see Enabling Termination Protection for an Instance (p. 316).	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
VM Import	2010-11-15	Added information about VM Import, which allows you to import a virtual machine or volume into Amazon EC2. For more information, see Step 1: Install the Amazon EC2 CLI (p. 271).	15 December 2010
Basic monitoring for instances	2010-08-31	Added information about basic monitoring for EC2 instances.	12 December 2010
Cluster GPU instances	2010-08-31	Amazon EC2 offers cluster GPU instances (cg1.4xlarge) for high-performance computing (HPC) applications. For more information about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details.	14 November 2010
Filters and Tags	2010-08-31	Added information about listing, filtering, and tagging resources. For more information, see Listing and Filtering Your Resources (p. 606) and Tagging Your Amazon EC2 Resources (p. 610).	19 September 2010
Idempotent Instance Launch	2010-08-31	Added information about ensuring idempotency when running instances. For more information, see Ensuring Idempotency (p. 672).	19 September 2010
Micro instances	2010-06-15	Amazon EC2 offers the `t1.micro` instance type for certain types of applications. For more information, see T1 Micro Instances (p. 118).	8 September 2010
AWS Identity and Access Management for Amazon EC2		Amazon EC2 now integrates with AWS Identity and Access Management (IAM). For more information, see Controlling Access to Amazon EC2 Resources (p. 429).	2 September 2010
Cluster instances	2010-06-15	Amazon EC2 offers cluster compute instances for high-performance computing (HPC) applications. For more information about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details.	12 July 2010

Feature	API Version	Description	Release Date
Amazon VPC IP Address Designation	2010-06-15	Amazon VPC users can now specify the IP address to assign an instance launched in a VPC.	12 July 2010
Amazon CloudWatch Monitoring for Amazon EBS Volumes		Amazon CloudWatch monitoring is now automatically available for Amazon EBS volumes. For more information, see Monitoring Volumes with CloudWatch (p. 534).	14 June 2010
High-memory extra large instances	2009-11-30	Amazon EC2 now supports a High-Memory Extra Large (m2.xlarge) instance type. For more information about the hardware specifications for each Amazon EC2 instance type, see Instance Type Details.	22 February 2010

AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the *AWS General Reference*.