Amazon Elastic Compute Cloud

Getting Started Guide

API Version 2011-12-15

Amazon Web Services

Amazon Elastic Compute Cloud: Getting Started Guide

Amazon Web Services

Getting Started with Amazon EC2 1
Get Started with EC2 with a Linux/UNIX Instance 2
Sign Up for EC2 3
Launch an Instance 4
Connect to Your Linux/UNIX Instance 9
Terminate Your Instance 19
Where Do I Go from Here? 21
Please Provide Feedback 26
Document History 27
About This Guide 28

Getting Started with Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that enables you to launch and manage Linux/UNIX and Windows server instances in Amazon's data centers. You can get started with Amazon EC2 by following the tasks shown in the following diagram. You'll primarily use the AWS Management Console, a point-and-click web-based interface.

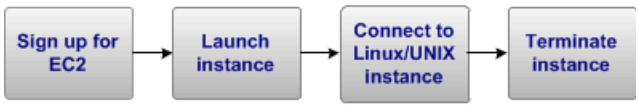
What Would You Like to Do?

- Get Started with EC2 with a Linux/UNIX Instance (p. 2)
- Get Started with EC2 with a Windows Instance

Get Started with EC2 with a Linux/UNIX Instance

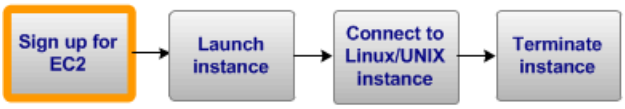
You can get started with Amazon EC2 and using a Linux/UNIX instance by following the tasks shown in the following diagram. You'll primarily use the AWS Management Console, a point-and-click web-based interface.

This document focuses on launching and connecting to an Amazon EC2 Linux/UNIX server instance. To get started with a Windows instance, go to Getting Started in the *Amazon EC2 Windows User's Guide*.



This guide walks you through launching and connecting to your first Amazon EC2 instance.

Sign Up for EC2



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

Note

When you create an account, AWS automatically signs up the account for all services. You are charged only for the services you use.

To create an AWS account

1. Go to http://aws.amazon.com, and click **Create an AWS Account**.
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.

Launch an Instance

Now that you're signed up for Amazon EC2, you're ready to launch an instance using the AWS Management Console.

You can either leverage the Free Usage Tier to launch and use a free Amazon EC2 Micro Instance for 12 months, or you can launch a regular instance (not within the Free Usage Tier). For more information about the Free Usage Tier, go to the AWS Free Usage Tier product page and Getting Started with AWS Free Usage Tier.

If you want to launch a regular Linux/UNIX instance (not within the Free Usage Tier), you will incur the standard Amazon EC2 usage fees for the instance until this tutorial shows you how to terminate it in the last task. The total charges will be minimal (typically less than a few dollars). For more information about Amazon EC2 usage rates, go to the Amazon EC2 product page.

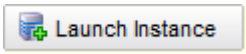
Important

If you launch an instance that is not within the Free Usage Tier, the usage fees are minimal, and you are billed once you launch the instance and charged for the time that the instance is running even if it remains idle.

To launch an instance

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

   Use the email address and password you used when signing up for Amazon EC2.
2. From the Amazon EC2 console dashboard, click **Launch Instance**.



The **Create a New Instance** page provides two ways to launch an instance:

- The **Classic Wizard** offers you more granular control and advanced settings for configuring the type of instance you want to launch.
- The **Quick Launch Wizard** simplifies the process for you and automatically configures many selections for you so you can started quickly with an instance. This tutorial guides you through the Quick Launch Wizard.

3. On the **Create a New Instance** page, click **Quick Launch Wizard**.
4. In **Name Your Instance**, enter an instance name that has meaning for you.
5. Under **Choose a Key Pair**, you can choose from any existing key pairs that you have created, or you can create a new one. For this example, we'll create a key pair:

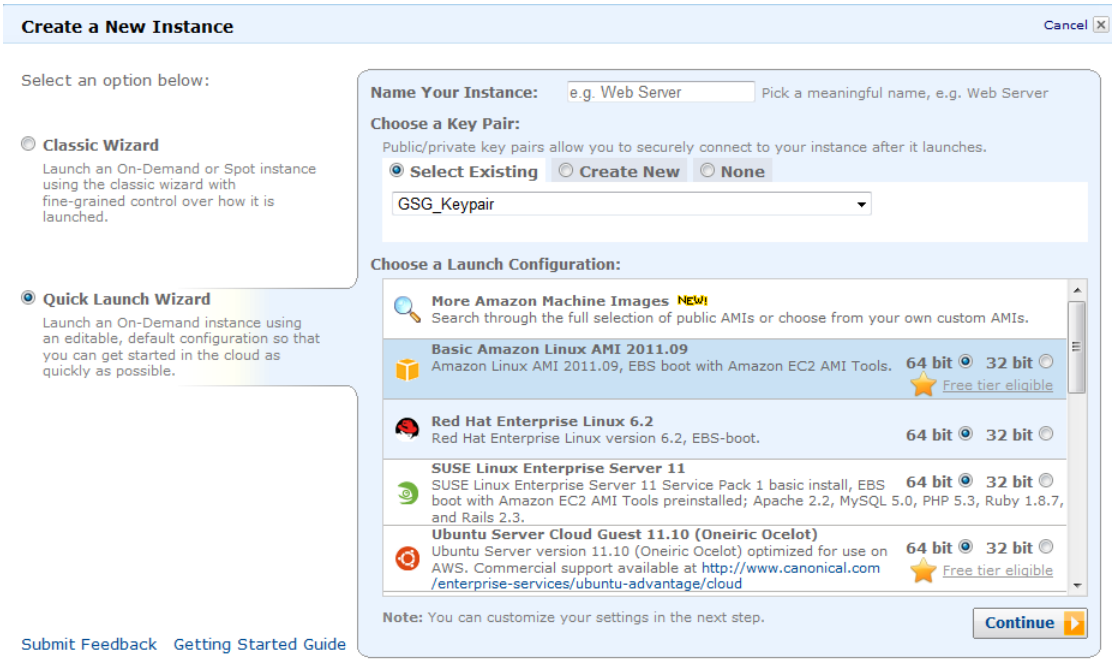
   Important

   Do not select the **None** option. If you launch an instance without a key pair, you will not be able to connect to your instance.

   a. Click **Create new Key Pair**.
   b. Type a name for your key pair and then click **Download**. You will need the contents of the private key to connect to your instance once it is launched. Amazon Web Services does not keep the private portion of key pairs.
   c. Save the private key in a safe place on your system. Note the location because you'll need the key to connect to the instance.

6. Under **Choose a Launch Configuration**, choose the operating system and software configuration for your instance. In this example, we'll use an Amazon Linux instance with a 64-bit operating system. The star by this choice indicates that it is within the Free Usage Tier.

   The Quick Launch Wizard displays a list of basic configurations called Amazon Machine Images (AMIs) that you can choose from to launch an instance. An Amazon Machine Image (AMI) contains all the information needed to create a new instance of a server. For example, an AMI might contain all the software to act as a web server (e.g., Linux, Apache, and your website). To keep things simple, AWS marks the AMIs that are available in the Free Usage Tier with a star.



7. Click **Continue** to view the settings that your instance will launch with.
8. Under **Security Details**, in **Security Group**, the wizard automatically makes a security group selection for you.

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

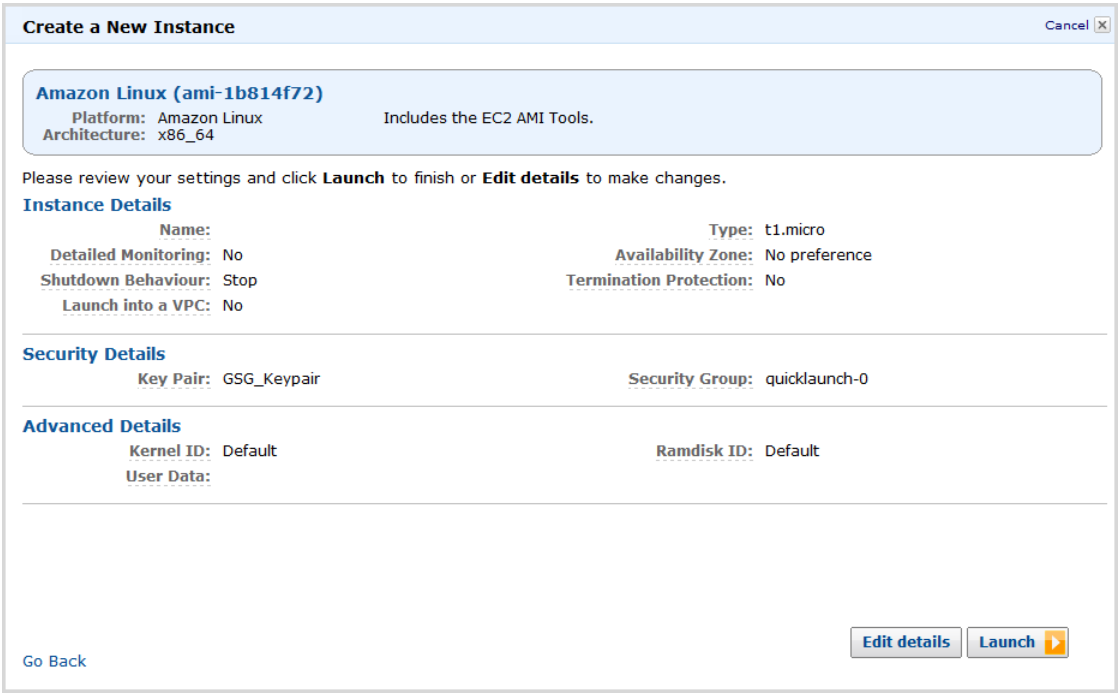
   If you're new to Amazon EC2 and haven't set up any security groups yet, AWS defines a default security group for you. The name and description for the group is quicklaunch-*x* where *x* is a number associated with your quicklaunch group. The first security group you create using the Quick Launch Wizard is named quicklaunch-1. You can change the name and description using the **Edit details** button. For example, in this tutorial we changed the name to quicklaunch-0. The group already has basic firewall rules that enable you to connect to the type of instance you choose. For a Linux instance, you connect through SSH on port 22. The quicklaunch-*x* security group automatically allows SSH traffic on port 22.

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

   Caution

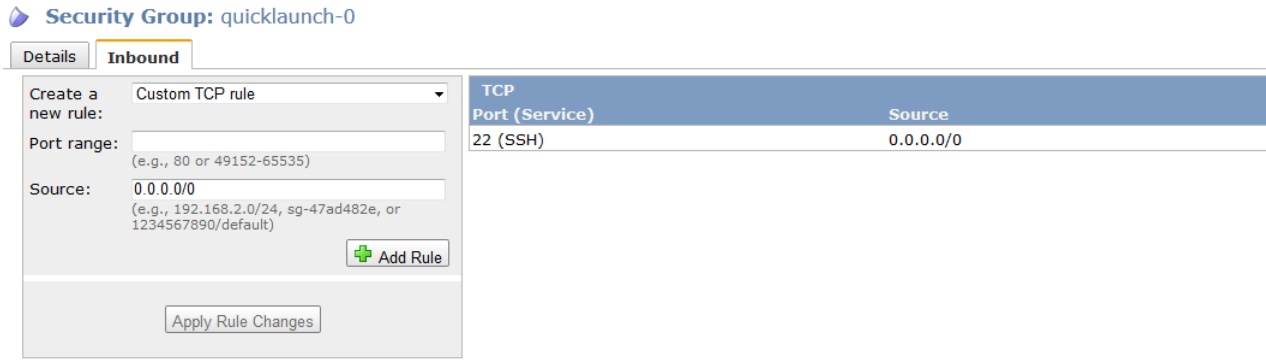
   The quicklaunch-*x* security group enables all IP addresses to access your instance over the specified ports (e.g., SSH for Linux/UNIX). This is acceptable for the short exercise in this tutorial, but it's unsafe for production environments. In production, you'll authorize only a specific IP address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



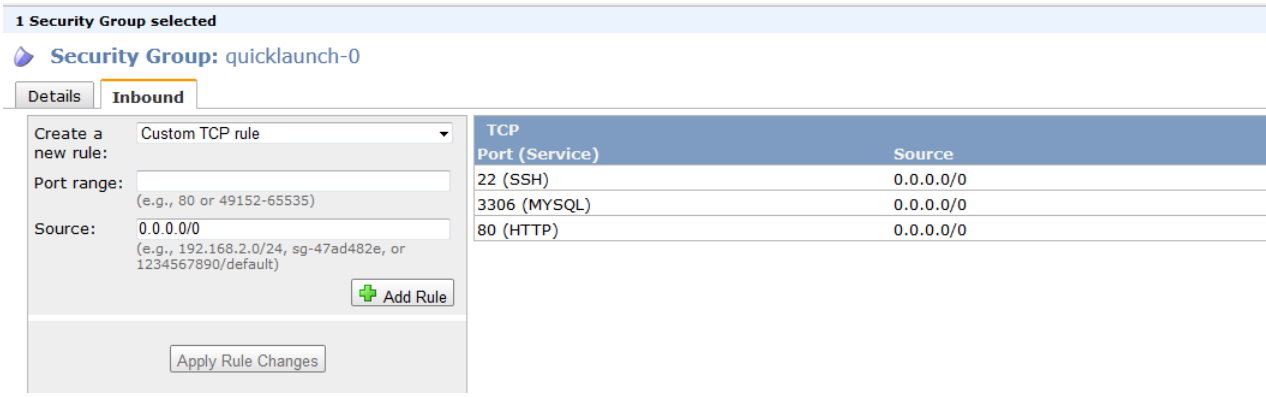
9. Review your settings, and click **Launch** to launch the instance.
10. A confirmation page lets you know your instance is launching. Click **Close** to close the confirmation page.
11. In the **Navigation** pane, click **Instances** to view the status of your instance. It takes a short time for an instance to launch. The instance's status will be *pending* while it's launching.

After a short period, the status of your instance switches to *running*. You can click **Refresh** to refresh the display.

12. Record the **Public DNS** name for your instance because you'll need it for the next task. If you select the instance, its details (including the public DNS name) are displayed in the lower pane. You can also click **Show/Hide** in the top right corner of the page to select which columns to display.
13. (Optional) After your instance is launched, you can view the quicklaunch-*x* security group rule that was created.
    a. On the Amazon EC2 console, under **Network and Security**, click **Security Groups**.
    b. Click the quicklaunch-*x* security group and you can view the security rules created.

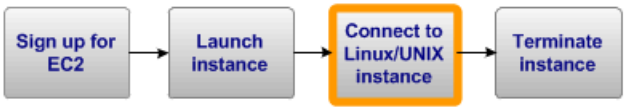


As you can see, the security group contains one rule that allows SSH traffic from any IP source to port 22. If you had launched a Linux instance with Apache and MySQL installed, the Quick Launch wizard would create a security group that would also allow traffic to port 80 for HTTP (for web traffic) and port 3306 for MySQL, as shown in the following figure.



When your instance's status is *running*, you can connect to it. To connect to your Linux/UNIX instance, see Connect to Your Linux/UNIX Instance (p. 9).

Connect to Your Linux/UNIX Instance



Topics

- Connecting from Your Web Browser Using the MindTerm SSH Client (p. 10)
- Connecting from a Linux/UNIX Machine Using a Standalone SSH Client (p. 11)
- Connecting from a Windows Machine Using PuTTY (p. 14)

Connecting from Your Web Browser Using the MindTerm SSH Client

The steps to connect to a Linux/UNIX instance using your browser are as follows:

1. Install and Enable Java on Your Browser (p. 10)
2. Connect Using the MindTerm (SSH) Client (p. 10)

Install and Enable Java on Your Browser

To connect to your instance from the Amazon Elastic Compute Cloud (Amazon EC2) console, you must have Java installed and enabled in your browser. To install and enable Java, follow the steps Oracle provides below or contact your IT administrator to install and enable Java on your web browser.

1. Install Java (see http://java.com/en/download/help/index_installing.xml).
2. Enable Java in your web browser (see http://java.com/en/download/help/enable_browser.xml).

Connect Using the MindTerm (SSH) Client

To connect to your instance through a web browser

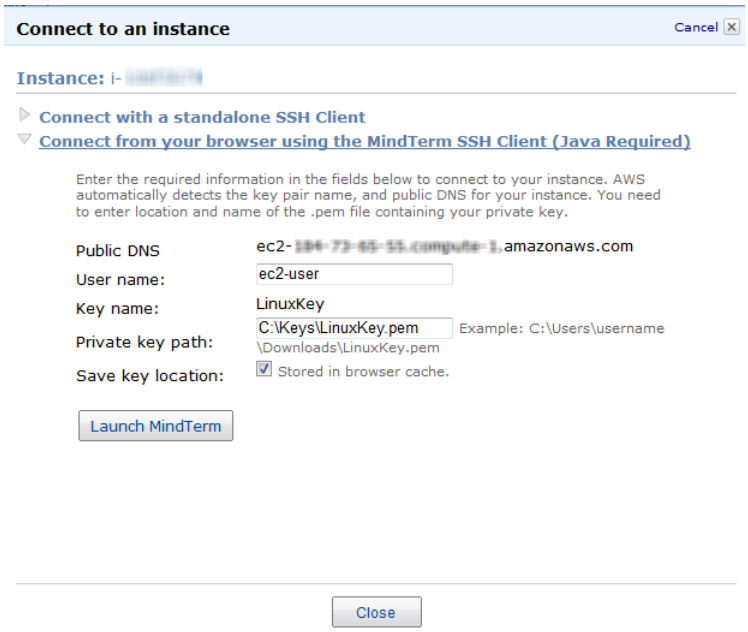
1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the **Navigation** pane, click **Instances**.
3. Right-click your instance, and then click **Connect**.
4. Click **Connect from your browser using the MindTerm SSH client (Java Required)**.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

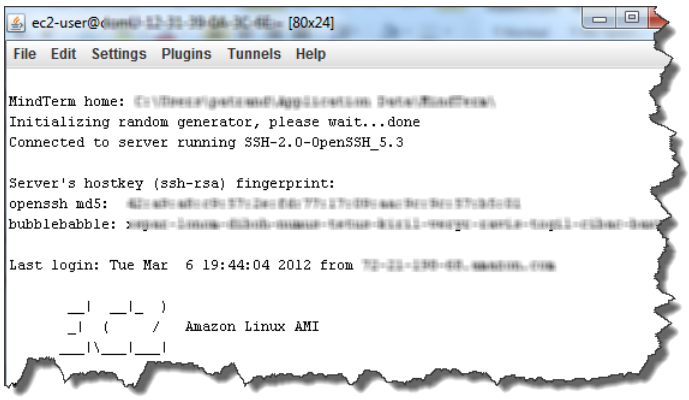
   Note

   For an Amazon Linux instance, the default user name is `ec2user`. For Ubuntu, the default user name is `ubuntu`. Some AMIs allow you to log in as root.

6. The **Key name** field is automatically populated for you.
7. In **Path to private key**, enter the fully qualified path to your .pem private key file.
8. Click **Save key location**, click **Stored in browser cache** to store the key location in your browser cache so the key location is detected in subsequent browser sessions, until your clear your browser's cache.
9. Click **Launch MindTerm**.



10. If necessary, click **Yes** to trust the certificate.
11. Click **Run** to run the MindTerm client.
12. If you accept the license agreement, click **Accept**.
13. If this is your first time running MindTerm, a series of dialog boxes will ask you to confirm setup for your home directory and other settings.
14. Confirm settings for MindTerm setup.
15. A screen similar to the following opens and you are connected to your instance.



Connecting from a Linux/UNIX Machine Using a Standalone SSH Client

Use the ssh command to connect to your Linux/UNIX instance from a Linux/UNIX machine.

Note

Most Linux and UNIX machines include an SSH client by default. If yours doesn't, the OpenSSH project provides a free implementation of the full suite of SSH tools. For more information, go to http://www.openssh.org.

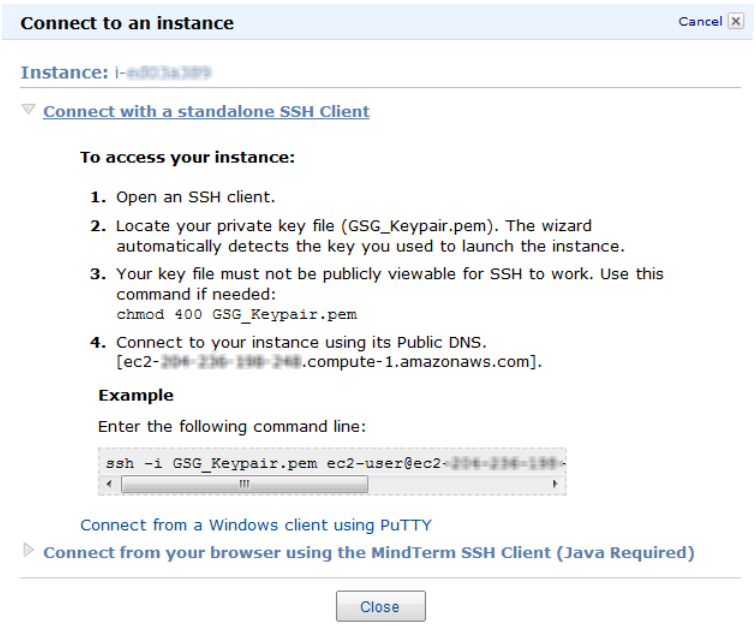
To use SSH to connect

1. On your Linux computer, use the `chmod` command to make sure your private key file isn't publicly viewable. For example, if your file were `GSG_Keypair.pem`, you would enter the following:

```
chmod 400 GSG_Keypair.pem
```

2. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
3. In the **Navigation** pane, click **Instances**.
4. Right-click your instance, and then click **Connect**.
5. Click **Connect using a standalone SSH client**. AWS automatically detects the public DNS address of your instance and the key pair name you launched the instance with.
6. Copy the example command provided in the Amazon EC2 console if you launched an Amazon Linux instance. If you used a different Amazon Machine Image (AMI) for your Linux/UNIX instance, you need to log in as the default user for the AMI. For an Ubuntu instance, the default user name is `ubuntu`. Some AMIs allow you to log in as `root` so you will need to change the user name from `ec2user` to the appropriate user name.

```
ssh -i GSG_Keypair.pem ec2-user@ec2-184-72-204-112.compute-1.amazonaws.com
```



You'll see a response like the following.

```
The authenticity of host 'ec2-184-72-204-112.compute-1.amazonaws.com
(10.254.142.33)'
can't be established.
RSA key fingerprint is 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00.
Are you sure you want to continue connecting (yes/no)? yes
```

7. Enter **yes**.

   You'll see a response like the following.

```
Warning: Permanently added 'ec2-184-72-204-112.compute-1.amazonaws.com'
(RSA)
to the list of known hosts.
```

Note

If you can't connect, check that SSH traffic is enabled for your instance. For more information, go to Authorize Network Access to Your Instances in the *Amazon Elastic Compute Cloud User Guide*.

You're now logged in as ec2-user and can work with the instance as you would any typical server. If you need to run a command as root, you must prefix the command with `sudo`. For example:

```
sudo /bin/cat /etc/image-id
```

Normally you'd continue using the instance. However, for the purposes of this tutorial, we're going to show you how to terminate the instance immediately. Jump to Terminate Your Instance (p. 19).

Connecting from a Windows Machine Using PuTTY

To connect to your Linux/UNIX instance from a Windows machine, you use an SSH client. The following instructions assume that you're using PuTTY, a free SSH client for Windows machines.

Getting PuTTY

To download and install PuTTY

- Go to http://www.chiark.greenend.org.uk/~sgtatham/putty/ and follow the instructions there.

Note

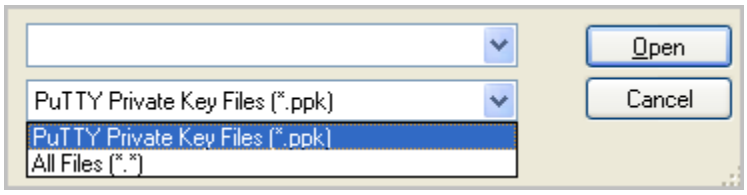
Other tools in the PuTTY suite are PuTTYgen, a key generation program, and pscp, a secure copy command line tool. The different PuTTY tools are separate applications. You can install them separately or install the entire suite with a simple Windows installer. The following instructions assume you've installed the entire suite and can access all the components from the Windows Start menu.

Converting Your Private Key

PuTTY does not natively support the private key format generated by Amazon EC2. Fortunately, PuTTY has a tool called PuTTYgen, which can convert keys to the required PuTTY format.

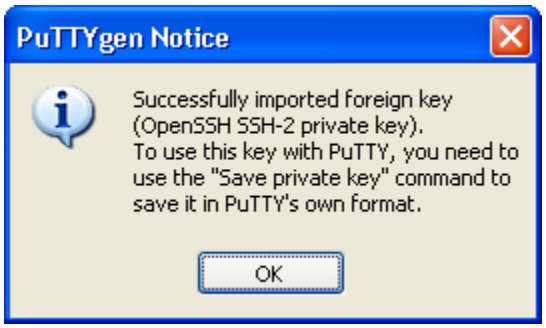
To convert your private key

1. Start PuTTYgen (e.g., from the **Start** menu, click **All Programs > PuTTY > PuTTYgen**).
2. Click **Load** and browse to the location of the private key file that you want to convert (e.g., `GSG_Keypair.pem`). By default, PuTTYgen displays only files with extension `.ppk`; you'll need to change that to display files of all types in order to see your `.pem` key file. The private key file must end with a newline character or PuTTYgen cannot load it correctly.



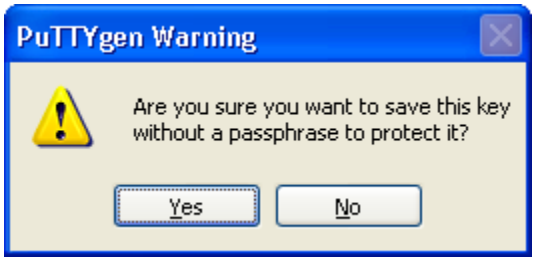
3. Select your `.pem` key file and click **Open**.

   PuTTYgen displays the following message.



When you click OK, PuTTYgen displays a dialog box with information about the key you loaded, such as the public key and the fingerprint. The keys that Amazon EC2 generates are 1024-bit SSH-2 RSA keys.

4. Click **Save private key** to save the key in PuTTY's format.
   PuTTYgen asks if you want to save the key without a passphrase.



5. Click **Yes**.

   Note

   A passphrase on a private key is an extra layer of protection, so even if your private key is discovered, it will not be usable without the passphrase. The downside to using a passphrase is that it makes automation harder because human intervention is needed to log on to an instance, or copy files to an instance. For this exercise, we're not using a passphrase.

6. Name the key with the same name you used for the key pair (e.g., GSG_Keypair). PuTTY automatically adds the `.ppk` file extension.

Your private key is now in the correct format for use with PuTTY. You can now connect to your instance using PuTTY's SSH client.

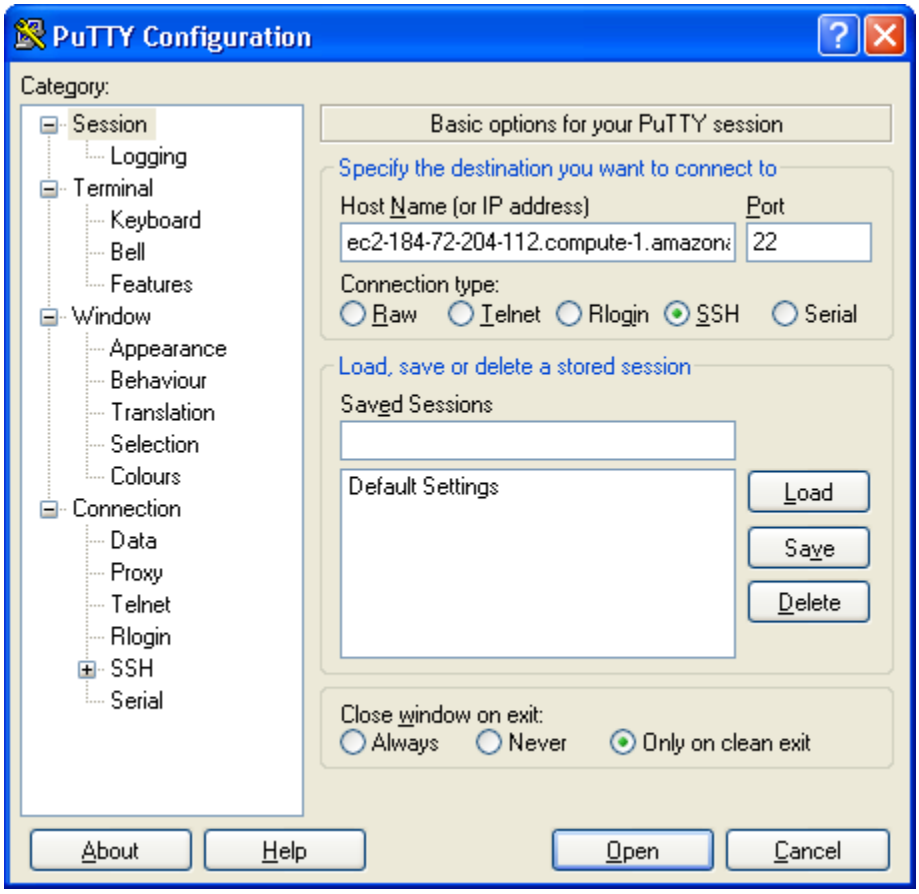
Connecting Using PuTTY SSH

You'll connect by starting a PuTTY SSH session.

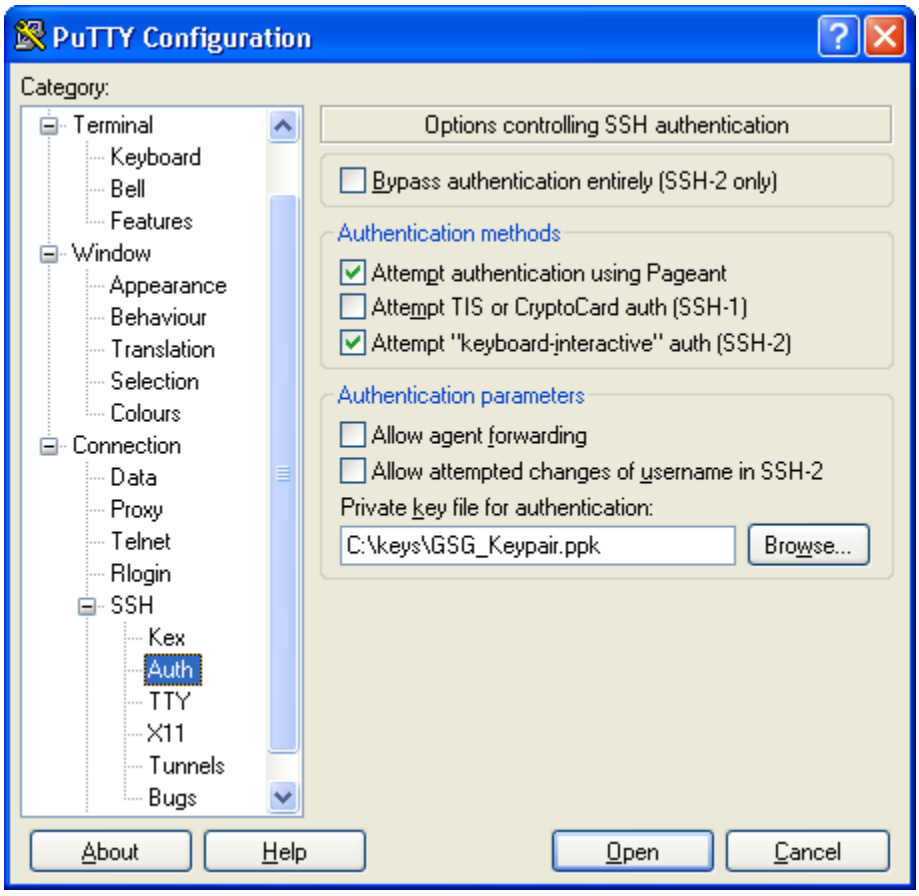
To use SSH to connect

1. Start PuTTY (e.g., from the **Start** menu, click **All Programs, PuTTY, PuTTY**).
   A dialog box opens with a **Category** menu on the left side. On the right side, the basic options for your PuTTY session are displayed.
2. In the **Host Name** field, enter the public DNS name of your instance (which you should have recorded earlier). If you launched an Amazon Linux instance, you can optionally prefix the DNS name with `ec2-user@` to automatically log in as ec2-user when the session opens. If you used a different

Amazon Machine Image (AMI) for your Linux/UNIX instance, you need to log in as the default user for the AMI. For an Ubuntu instance, the default user name is `ubuntu`. Some AMIs allow you to log in as `root`.



3. In the **Category** menu, under **Connection**, click **SSH**, and then **Auth**.
   The options controlling SSH authentication are displayed.
4. Click **Browse** and navigate to the PuTTY private key file you generated in the preceding section.



5. Click **Open**.
   An SSH session window opens and PuTTY displays a security alert asking if you trust the host you're connecting to.
6. Click **Yes**.

   Note

   If you can't connect, check that SSH traffic is enabled for your instance. For more information, go to Authorize Network Access to Your Instances in the *Amazon Elastic Compute Cloud User Guide*.

7. In the SSH session window, log in as ec2-user if you didn't as part of starting the SSH session.

   Tip

   The AMI we launched in this exercise requires you to log in to your instance as ec2-user. Some AMIs let you log in as root.

   Note

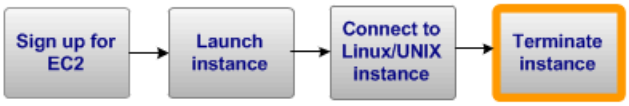
   If you specified a passphrase when you converted your private key to PuTTY's format, you must provide that passphrase when you log in to the instance.

You're now logged in as ec2-user and can work with the instance like you would any normal server. If you need to run a command as root, you must prefix the command with `sudo`. For example:

```
sudo /bin/cat /etc/image-id
```

Normally you'd continue using the instance. However, for the purposes of this guide, we're going to show you how to terminate the instance immediately. Jump to Terminate Your Instance (p. 19).

Terminate Your Instance



If the instance you launched was not in the free usage tier, as soon as your instance starts to boot, you're billed for each hour or partial hour that you keep the instance running (even if the instance is idle). When you've decided that you no longer need the instance, you can terminate it.

For more information about the free usage tier, go to the AWS Free Usage Tier product page and Getting Started with AWS Free Usage Tier.

Note

You cannot restart a terminated instance. However, you can launch additional instances of the same AMI.

To terminate an instance

1. In the AWS Management Console, locate the instance in your list of instances on the **Instances** page.
2. Right-click the instance, and then click **Terminate**.
3. Click **Yes, Terminate** when prompted for confirmation.
   Amazon EC2 begins terminating the instance. As soon as the instance status changes to `shutting down` or `terminated`, you stop incurring charges for that instance.

Congratulations! You successfully launched, connected to, and terminated an instance. For more information about Amazon EC2 and how to continue, see Where Do I Go from Here? (p. 21).

Your input is important to us. Help make our documentation helpful and easy to use. Please take a minute to provide feedback on your getting started experience with Amazon EC2. To begin the survey, see Please Provide Feedback (p. 26). Thank you.

Where Do I Go from Here?

Topics

- AWS Account and Security Credentials (p. 21)
- Designing Your Application for the Cloud (p. 21)
- Learn More about Amazon EC2 (p. 22)
- Amazon EC2 Resources (p. 24)

Amazon EC2 is a rich service offering many things we haven't covered in this guide, such as creating your own AMIs, using persistent file storage, monitoring instance health, load balancing, and virtual private networking. This section provides links to additional resources, which will help you deepen your understanding and use of Amazon EC2.

AWS Account and Security Credentials

So far you signed up for the service, got an AWS account and security credentials, and then completed a short exercise covering the essential product functions. Now that you're finished with the exercise, we recommend that you check with an administrator or coworker in your organization to determine if he or she already has an AWS account and security credentials for you to use in future interactions with AWS.

If you're an account owner or administrator and want to know more about AWS Identity and Access Management, go to the product description at http://aws.amazon.com/iam or to the technical documentation at Using AWS Identity and Access Management.

Designing Your Application for the Cloud

AWS solutions architects and evangelists have written white papers to help you design your application so it's fault tolerant, scalable, and elastic. For more information, go to AWS Cloud Computing Whitepapers.

Learn More about Amazon EC2

This section lists additional features of Amazon EC2 and where to get more information. You can also find additional information about Amazon EC2 in the Amazon EC2 Articles & Tutorials area of the AWS web site.

Amazon Virtual Private Cloud

You can use Amazon EC2 with Amazon Virtual Private Cloud, a service that enables you to create an isolated portion of the AWS cloud called a *VPC*. With Amazon VPC, you can create a virtual network topology—including subnets and route tables—for your EC2 resources. For more information, go to the Amazon VPC product page and the Amazon Virtual Private Cloud User Guide.

Creating Your Own AMIs

Amazon and other reputable sources offer AMIs that you can launch. However, you might want to create your own custom AMIs. You can modify instances of Amazon AMIs or other reputable public AMIs as needed and create your own custom AMIs from them. For general information about AMIs, go to AMIs and to Creating Your Own AMIs in the *Amazon Elastic Compute Cloud User Guide*.

You can choose between Amazon S3 or Amazon Elastic Block Store as the root device for your AMI (for a brief description of Amazon EBS, see Amazon Elastic Block Store (p. 23) later in this section). We recommend using instances backed by Amazon EBS, because they launch faster and use persistent storage. For more information, go to AMIs Backed by Amazon EBS in the *Amazon Elastic Compute Cloud User Guide*.

Importing Your Own Virtual Machines

You can import a virtual machine or volume from your own data center into Amazon EC2. For more information, go to Importing Your Virtual Machines and Volumes into Amazon EC2 in the *Amazon Elastic Compute Cloud User Guide*.

Instance Types

To meet the needs of different organizations and applications, Amazon EC2 instances are available in different sizes and CPU/memory configurations. For more information, go to Instances in the *Amazon Elastic Compute Cloud User Guide*.

Tags

You can add optional metadata to your instances, AMIs, and other EC2 resources to help you categorize and manage them. For more information, go to Using Tags in the *Amazon Elastic Compute Cloud User Guide*.

Elastic IP Addresses

You might want to have static IP addresses for your instances. Amazon EC2 provides *elastic IP addresses* that can be dynamically remapped to different instances. For more information, go to Elastic IP Addresses in the *Amazon Elastic Compute Cloud User Guide*.

Security Groups

You might be concerned about keeping others from accessing your instances, both inside and outside the Amazon network. You can create other security groups (beyond the basic group we used in this guide) to meet your security requirements. For more information, go to Network Security Concepts in the *Amazon Elastic Compute Cloud User Guide*.

Availability Zones

You might want to build a geographically dispersed, fault tolerant architecture on Amazon EC2. You can place instances in different geographic regions and isolate instances within those regions using Availability Zones. This provides geographic flexibility and affordable fault tolerance. For more information, go to Region and Availability Zone Concepts in the *Amazon Elastic Compute Cloud User Guide*.

Amazon Linux

AWS provides Amazon Linux AMIs, which are supported and maintained Linux images optimized for the EC2 environment. For more information, go to Amazon Linux AMI.

Amazon EC2 Running Windows

Amazon EC2 can run Microsoft Windows Server, with or without Microsoft SQL Server. For more information, go to the Amazon EC2 Running Microsoft Windows Server and SQL Server page. Also, go to Instance Families and Types and look for Windows Instance Types in the *Amazon Elastic Compute Cloud User Guide*.

Reserved Instances

You might want to run a set of full-time or nearly full-time instances but also bring down your costs. Amazon EC2 supports an additional pricing option that enables you to make a low one-time payment for each instance to reserve and receive a significant discount on the hourly usage charge for that instance. For more information, go to On-Demand and Reserved Instances and to Reserving Amazon EC2 Instances in the *Amazon Elastic Compute Cloud User Guide*.

Spot Instances

If you're flexible about when you need instances and want to bring down your costs, Amazon EC2 lets you bid for unused Amazon EC2 capacity and run your instances for as long as your bid exceeds the current *Spot Price*. For more information, go to the Amazon EC2 Spot Instances product page and Introduction to Spot Instances.

Amazon Elastic Block Store

You might need more space than is provided on the instance, or you might need a permanent storage solution. Amazon Elastic Block Store enables you to create volumes that can be mounted as block devices by Amazon EC2 instances. Amazon EBS volumes behave like raw unformatted external block devices, and they persist past the life of an Amazon EC2 instance. For more information, go to the Amazon Elastic Block Store product page. Also go to Amazon Elastic Block Store in the *Amazon Elastic Compute Cloud User Guide*.

Monitoring Instances

You might need a solution for monitoring your instances. Amazon CloudWatch is a monitoring service for Amazon EC2 that is designed to gather, aggregate, store, and retrieve metrics. For more information, go to the Amazon CloudWatch product page and the Amazon CloudWatch Developer Guide.

Load Balancing

You might need a solution for load balancing requests to your instances. Elastic Load Balancing offers the ability to evenly spread requests across your running Amazon EC2 instances. For more information, go to the Elastic Load Balancing product page and the Elastic Load Balancing Developer Guide.

Automatically Scaling Instances

You might want to automatically scale up and down the number of instances you use. Auto Scaling enables you to automatically increase or decrease the number of running Amazon EC2 instances in response to your web application's usage and the configuration you define. For more information, go to the Auto Scaling product page and the Amazon Auto Scaling Developer Guide.

Micro Instances

Amazon EC2 offers micro instances, which provide a small amount of consistent CPU resources and allow you to burst CPU capacity when additional cycles are available. They are well suited for lower throughput applications and web sites that consume significant compute cycles periodically. For more information, go to Micro Instance Concepts in the *Amazon Elastic Compute Cloud User Guide*.

Cluster Instances

Amazon EC2 offers cluster instances for your High-Performance Computing (HPC) applications. These instances provide you with high-bandwidth, low-latency inter-node communications for advanced computational applications such as computational fluid dynamics, computational biology, and materials research. For more information, go to Using Cluster Instances in the *Amazon Elastic Compute Cloud User Guide*.

Public Data Sets

Amazon EC2 provides a repository of public data sets, such as the mapping of the human genome and the US census data, that you can seamlessly integrate into your AWS cloud-based applications. For more information, go to the Public Data Sets on AWS page. Also go to Using Public Data Sets in the *Amazon Elastic Compute Cloud User Guide*.

Amazon EC2 Resources

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

Resource	Description
Amazon Elastic Compute Cloud User Guide	Provides conceptual information about Amazon EC2 and describes how to use Amazon EC2 features using the AWS Management Console, command line tools, and Query API.

Resource	Description
Amazon Elastic Compute Cloud API Reference	Contains a comprehensive description of the API actions, data types, and errors.
Amazon Elastic Compute Cloud Command Line Reference	Contains a comprehensive description of all the command line tools and their options.
Amazon EC2 Technical FAQ	Covers the top questions developers have asked about this product.
Amazon EC2 Release Notes	Give a high-level overview of the current release. They specifically note any new features, corrections, and known issues.
AWS Developer Resource Center	A central starting point to find documentation, code samples, release notes, and other information to help you build innovative applications with AWS.
AWS Management Console	The console lets you perform most of the functions of Amazon EC2 and other AWS products without programming.
Discussion Forums	A community-based forum for developers to discuss technical questions related to Amazon Web Services.
AWS Support Center	The home page for AWS Technical Support, including access to our Developer Forums, Technical FAQs, Service Status page, and AWS Premium Support (if you are subscribed to this program).
AWS Premium Support Information	The primary web page for information about AWS Premium Support, a one-on-one, fast-response support channel to help you build and run applications on AWS Infrastructure Services.
Amazon EC2 Product Information	The primary web page for information about Amazon EC2.
Form for questions related to your AWS account: Contact Us	This form is *only* for account questions. For technical questions, use the Discussion Forums.
Terms of Use	Detailed information about the copyright and trademark usage at Amazon.com and other topics.

Your input is important to us. Help make our documentation helpful and easy to use. Please take a minute to provide feedback on your getting started experience with Amazon EC2. To begin the survey, see Please Provide Feedback (p. 26). Thank you.

Please Provide Feedback

Your input is important to help make our documentation helpful and easy to use. Please tell us about your experience getting started with Amazon EC2 by completing our Getting Started Survey.

Thank you.

Document History

This documentation is associated with the 2011-12-15 release of Amazon Elastic Compute Cloud (Amazon EC2). This guide was last updated on 07 March 2012.

The following table describes the important changes since the last release of the *Amazon Elastic Compute Cloud Getting Started Guide*.

Change	Description	Release Date
Public Release	This is the first release of the *Amazon Elastic Compute Cloud Getting Started Guide*.	In this release

About This Guide

This is the *Amazon Elastic Compute Cloud Getting Started Guide*. It was last updated on March 07, 2012.

Amazon Elastic Compute Cloud is often referred to within this guide as "Amazon EC2" or simply "EC2"; likewise the Amazon Simple Storage Service is referred to in this guide as "Amazon S3"; all copyrights and legal protections still apply.