Amazon Elastic Compute Cloud

Getting Started Guide

API Version 2009-04-04

Amazon Elastic Compute Cloud: Getting Started Guide

Table of Contents

Welcome

Topics

This is the *Amazon Elastic Compute Cloud* Getting Started Guide. This section describes who should read this guide, how the guide is organized, and other resources related to Amazon EC2.

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.

Who Should Read This Guide

This guide is intended for developers and system administrators who need compute capacity that they can scale up and down within minutes, not hours or days.

Required Knowledge and Skills

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

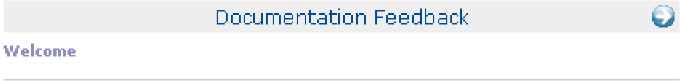
- Basic understanding of web services (Go to the *W3 Schools Web Services Tutorial*)

Note

Although you can use the AWS Management Console for most tasks in this guide, we recommend installing the Amazon EC2 command line tools to access some features that might not be available on the console (Go to the *Amazon EC2 Command Line Tools Page*).

Reader Feedback

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



Documentation Feedback

Welcome

How to Use this Guide

This guide is organized as a high-level introduction and tutorial. It is divided into several major sections that allow you to practice using Amazon EC2 in a simple environment. This guide provides you with informatin on how to set up your account, describes how to launch instances using the AWS Management Console or the command line tools, and provides next steps that describe major features that you might want to use.

Paths through this Guide

To determine how to use this guide, select from the following:

- **I'm a newbie—**If you are a beginning user or just want to quickly try out Amazon EC2, then set up your account (AWS, Amazon EC2, and Amazon S3) and use the AWS Management Console. For more information, see Setting up an Account (p. 8) and Getting Started with the Console (p. 11).
- **I need to understand the APIs—**The fastest way to get a feeling for the APIs without setting up your environment is to use the command line tools. They are very similar to the APIs and easy to set up. For more information, see Setting up an Account (p. 8) and Getting Started with the Command Line Tools (p. 18).
- **I need to understand Amazon EC2, not play with it—**For conceptual information about Amazon EC2, go to the Amazon Elastic Compute Cloud User Guide.
- **This is old hat; get me out of here—**If you are already familiar with Amazon EC2 and have set up your environment, see Where Do I Go from Here? (p. 45) for brief information about major Amazon EC2 features that you might want to use. Also, go to the Amazon Elastic Compute Cloud User Guide or Amazon Elastic Compute Cloud Developer Guide, depending on whether you are an Amazon EC2 console user or a developer.

Sections of this Guide

The major sections of this guide are:

- Introduction to Amazon Elastic Compute Cloud (p. 6)
- Setting up an Account (p. 8)
- Getting Started with the Console (p. 11)
- Getting Started with the Command Line Tools (p. 18)
- Where Do I Go from Here? (p. 45)
- Appendix: PuTTY (p. 47)
- Document Conventions (p. 51)

Amazon EC2 Resources

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

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

What's New

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

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

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

Change	Description	Release Date
Amazon CloudWatch	Amazon CloudWatch is a monitoring service for Amazon EC2 that is designed to gather, aggregate, store, and retrieve metrics. Amazon CloudWatch makes it easy to monitor your Amazon EC2 instances and aggregate metrics from instances like CPU or disk utilization over different time ranges and across different pools of resources. This service is tightly integrated with Amazon EC2's Auto Scaling and Elastic Load Balancing, enabling you to use monitoring metrics to trigger scaling activities. For more information, see Amazon CloudWatch Developer Guide.	18 May 2009
New Guides	Amazon EC2 now consists of six guides: • **Amazon Elastic Compute Cloud Getting Started Guide**—Describes how to set up your environment and get started with Amazon EC2. • **Amazon Elastic Compute Cloud User Guide**—Describes Amazon EC2 concepts and how to use Amazon EC2 with the AWS Management Console or the command line tools. • **Amazon Elastic Compute Cloud Developer Guide**—Describes Amazon EC2 concepts and how to use Amazon EC2 with the APIs. • **Amazon Elastic Compute Cloud API Reference**—Provides detailed information about the Amazon EC2 APIs. • **Amazon Elastic Compute Cloud Command Line Reference**—Provides detailed information about the Amazon EC2 command line tools.. • **Amazon Elastic Compute Cloud Quick Reference Card**—Provides a quick summary of the Amazon EC2 command line tools.	18 May 2009

Introduction to Amazon Elastic Compute Cloud

Topics

What is Amazon EC2?

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

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

Popular Uses for Amazon EC2

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

- **Scalable Applications—**You can build a scalable application that shrinks or expands to meet your current demands.
  This can help you use only the compute resources that you need and can help you respond to events where a mention on a popular news site can result in a dramatic spike in traffic.
- **Temporary Events—**You can use Amazon EC2 for temporary solutions and one-off events that would require you to maintain a fleet of compute resources that are normally idle.
  This includes hosting conferences in virtual worlds, live blogging, distribution of newly released media, and short-term promotional web sites.

- **Batch Processing—**You can use Amazon EC2 for projects that require massive compute resources which would be expensive to build on your own.
  This includes video and image processing, financial data processing, and science and research applications.
- **Fault Resilient Applications—**You can build an application across multiple availability zones which will be protected against the loss of an entire physical location.

Setting up an Account

Topics

To use Amazon EC2, you must sign up for an AWS Account, sign up for Amazon Simple Storage Service (Amazon S3), and sign up for Amazon EC2. These are three different actions that must be performed separately.

After you sign up for these services, you will get your Access Key identifiers and get started with Amazon EC2 using the AWS Management Console or the command line tools.

Note

If you are already an Amazon S3 user, you can skip to Signing up for Amazon EC2 (p. 9)).

Signing up for an AWS account

This section describes how to sign up for an AWS account. If you already have an AWS account, you can skip this task.

Amazon S3 Signup Process

1	Go to the AWS home page.
2	Click the **Sign Up Now** button. If you don't already have an Amazon account, you are prompted to create one as part of the sign up process.

3	Follow the on-screen instructions.

Signing up for Amazon S3

Amazon EC2 AMIs are stored in and retrieved from Amazon S3. This means you need to sign up for Amazon S3. If you are already an Amazon S3 user, you can skip this task.

Amazon S3 Signup Process

1	Go to the Amazon S3 home page.
2	Click the **Sign up for this service** button. If you don't already have an AWS account, you are prompted to create one as part of the sign up process.
3	Follow the on-screen instructions.

After signing up for Amazon S3, point to the button labeled **Your Web Services Account** and select the **AWS Access Key Identifiers** link on the menu that appears. Be sure to note down your AWS account's Access Key ID and Secret Access Key. You need these to bundle your own image (see ???).

Signing up for Amazon EC2

After you sign up for Amazon S3, you'll need to sign up for Amazon EC2. If you are already an Amazon EC2 user, you can skip this task.

Amazon EC2 Signup Process

1	Log into your AWS account and follow the link to Amazon EC2 under the **Browse Web Services** section on the left.
2	Click **Sign Up For Web Service** in the top right of the screen and follow the on-screen instructions.
3	Follow the on-screen instructions.

Getting Your Access Key Identifiers

After you sign up for Amazon EC2, you need to get your Access Key identifiers if you want to access your account through the command line tools or the API. If you will only use the AWS Management Console, you can skip this task for now.

Access Key Identifiers Process

1	Point to the button labeled **Your Web Services Account** and select the **View Access Key Identifiers** link on the menu that appears.
2	Click **Create New** in the **Your X.509 Certificate** section to create a new X.509 certificate.
3	Save the certificate and private key. You'll need this when you set up our command line tools (see Setting up the Tools (p. 20)).

4	Create a `.ec2` directory in your home directory, and save these files to it with the filenames offered by your browser. You should end up with a PEM-encoded X.509 certificate and a private key file named as shown in the following examples.
	The following is an example of a PEM encoded signed X.509 certificate.
	`cert-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem`
	The following is an example of an unencrypted, PEM encoded RSA private key that corresponds to the preceding X.509 certificate.
	`pk-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem`
	This X.509 certificate is associated with your account until you generate or upload a new certificate. If you have an existing certificate that you prefer to use, you can return to the **Access Key Identifiers** upload it later.
	Note
	For Windows, there can be no spaces in the path. For example, `C:\EC2` is acceptable, but `C:\My Documents\EC2` is not.

5	Finally, you'll need to look up your AWS account ID. You should use this value whenever you need to provide an Amazon EC2 user ID. From the AWS portal page, point to **Your Web Services Account** and select the **Account Activity** link on the menu that appears. At the top of this page, locate your the **Account Number** which is a hyphenated number that looks similar to `4952-1993-3132`. This number, with the hyphens removed, is your AWS account ID. In this example, it is AIDADH4IGTRXXKCD.

Choosing Your Path

To determine how to proceed next, select from the following:

- **I'm a newbie—**If you are a beginning user or just want to quickly try out Amazon EC2, then use the AWS Management Console. For more information, see Getting Started with the Console (p. 11).
- **I need to understand the APIs—**The fastest way to get a feeling for the APIs without setting up your environment is to use the command line tools. They are very similar to the APIs and easy to set up. For more information, see Getting Started with the Command Line Tools (p. 18).
- **This is old hat; get me out of here—**If you are already familiar with Amazon EC2 and have set up your environment, see Where Do I Go from Here? (p. 45) for brief information about major Amazon EC2 features that you might want to use. Also, go to the Amazon Elastic Compute Cloud User Guide or Amazon Elastic Compute Cloud Developer Guide, depending on whether you are an Amazon EC2 console user or a developer.

Getting Started with the Console

Topics

The AWS Management Console is a user-friendly web-based application that enables you to launch and manage Amazon EC2 instances without using the APIs or command line tools.

This section describes how to access the console, the minimum steps you need to take to set up your Amazon EC2 environment, how to run an instance, how to access an instance, and how to terminate it.

Prerequisites

Topics

This document assumes that the reader is comfortable working in a Linux/UNIX or Windows environment.

Accessing Linux and UNIX Instances through SSH Clients

For some of the examples illustrated in this guide you'll need access to an SSH client. Most Linux and UNIX installations include an SSH client by default. If yours does not, the OpenSSH project provides a free implementation of the full suite of SSH tools. For more information, go to the http://www.openssh.org.

Windows users can download and install PuTTY, a free SSH client. To download the client and installation instructions, go to the http://www.chiark.greenend.org.uk/~sgtatham/putty/. For information on how to use PuTTY with Amazon EC2, see Appendix: PuTTY (p. 47).

Accessing Windows Instances through Remote Desktop Clients

Some of the examples in this guide require a Remote Desktop client. If you do not have one, go to the Microsoft home page.

How to Access the Console

This section describes how to access the AWS Management Console.

To access the console

1. Go to AWS Management Console.
2. Click the **Amazon EC2** tab.

   You are prompted to log in.
3. Enter the Amazon e-mail address and password associated with the account and click the **Sign In** button.

   The AWS Management Console opens, displaying the Amazon EC2 Console Dashboard page.

Note

If you are inactive, the AWS Management Console automatically logs you out. To disable this, click **Settings** and deselect the **Sign out on inactivity** check box.

How to Create a Key Pair

Before you begin, you must create a key pair. The key pair allows you to access your instance using SSH (Linux/UNIX) or RDP (Windows).

To generate a key pair

1. Log in to the AWS Management Console and click the **Amazon EC2** tab.
2. Click **Key Pairs** in the **Navigation** pane.

   The console displays a list of key pairs associated with your account.
3. Click **Create Key Pair**.

   The **Key Pair** dialog box appears.
4. Enter a name for the new key pair in the **Key Pair Name** field and click **Create**.

   You are prompted to download the key file.
5. Download the key file and keep it in a safe place. You will need it to access any instances that you launch with this key pair.

How to Add Rules to the Default Security Group

Before you launch an instance, you need to configure a security group. Security groups define access control permissions for instances. If you skip this task, you will not be able to access any instances that you launch.

In this example, we will add HTTP access on port 80, SSH access on port 22, and Remote Desktop (RDP) access on port 3389. This enables the instance to be reached on port 80 from the Internet and enables you to administer the instance over SSH or RDP.

Caution

In this example, you enable any IP address to access ports 22 and 3389 of the instance. Although this might be acceptable for testing purposes, it is extremely unsafe for production environments. For production systems, you must obtain your public IP address ranges and grant access to those ranges only. For example, if your IP address is 103.55.22.234, you specify `103.55.22.234/32`.

To add rules to the default security group

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

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

   Its rules appear in the lower pane.
4. To add the HTTP rule, enter the following:

   - Select `HTTP` from the **Connection Method** list box.
   - Select `TCP` from the **Protocol** list box.
   - Enter `80` in the **From Port** and **To Port** fields.
   - Enter `0.0.0.0/0` in the **Source** field.

   Then, click **Save**.
5. To add the SSH rule, enter the following:

   - Select `SSH` from the **Connection Method** list box.
   - Select `TCP` from the **Protocol** list box.
   - Enter `22` in the **From Port** and **To Port** fields.
   - Enter `0.0.0.0/0` in the **Source** field.

   Then, click **Save**.
6. To add the RDP rule, enter the following:

   - Select `RDP` from the **Connection Method** list box.
   - Select `TCP` from the **Protocol** list box.
   - Enter `3389` in the **From Port** and **To Port** fields.
   - Enter `0.0.0.0/0` in the **Source** field.

   Then, click **Save**.

How to Run an Instance

In this part of the tutorial, you select an AMI and launch it.

Note

To view available AMIs, click **AMIs** in the **Navigation** pane. Then, use the **Viewing** list boxes and fields to narrow your AMI choices.

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

To launch an instance of your AMI

1. Log in to the AWS Management Console and click the **Amazon EC2** tab.
2. Click **Instances** in the **Navigation** pane.

   The console displays a list of running instances.
3. Click **Launch Instance**.

   The Launch Instance wizard appears.
4. Select the **Quick Start** tab.
5. If you are launching a Linux/UNIX instance, locate the **Getting Started on Fedora Core 8** AMI and click its **Select** button. If you are launching a Windows instance, locate the **Getting Started on Microsoft Windows Server 2003** AMI and click its **Select** button.
6. If the **Configure Firewall** page of the wizard appears, click the **Skip** button because you already configured the `default` security group.

   The **Launch** page of the wizard appears.
7. Confirm the following settings and click **Launch**.

   - Enter `1` in the **Number of Instances** field.
   - Select the `m1.small`**Instance Type** option.
   - Select the key pair that you created from the **Key Pair Name** list box.
   - Select `default` from the **Security Groups** list box.

   The instance(s) begin launching.

Connecting to Your Instance

Topics

This part of the tutorial describes how to connect to your Linux/UNIX or Windows instance.

How to Connect to a Linux/UNIX Instance

To connect to a Linux/UNIX instance, you use the SSH command.

Note

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

To connect to your instance

1. Open a web browser and go to `http://<hostname>/`, where `<hostname>` is your instance's public hostname in the **Public DNS** field of the **Instances** page or as returned by ec2-describe-instances.

   A webpage welcoming you to your instance displays.

   Note

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

2. Whenever you launch a public AMI that you have not rebundled, locate the `SSH HOST KEY FINGERPRINTS` section of the instance output. You can get this information by selecting the instance and clicking **Output**.

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

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

3. Use the following command to login as root and exercise full control over this instance as you would any host. The private key that you downloaded when you created a key pair is used to connect to the host.

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

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

How to Connect to a Windows Instance

To connect to a Windows instance, you retrieve the initial administrator password and use Remote Desktop to connect.

To retrieve the initial administrator password

1. Log in to the AWS Management Console and click the **Amazon EC2** tab.
2. Click **Instances** in the **Navigation** pane.

The console displays a list of running instances.

3. Select the Windows instance, click **More Actions**, and select **Get Windows Password**.

   The **Retrieve Default Windows Administrator Password dialog** box appears.
4. Paste the contents of the Windows key pair file including the beginning and ending lines into the **Private Key** field and click **Decrypt Password**.

   The console returns the default administrator password for the instance.
5. Save the password. You will need it to log into the instance.

To connect to a Windows instance using Remote Desktop

1. Retrieve your instances public IP address from the **Public DNS** field of the **Instances** page of the AWS Management Console.
2. Log in to the instance using the RDP application of your choice.:

   Note

   For more information on the Windows Remote Desktop application, go to the Microsoft Web Site.
3. Enter the administrative password found in the previous section, using `Administrator` as the username.

How to Terminate an Instance

As soon as your instance starts to boot, you are billed for the resources it consumes. Once you have decided that you no longer need it, you can terminate an instance.

Note

You cannot recover a terminated instance. However, you can launch additional instances of an AMI.

To terminate an instance

1. Log in to the AWS Management Console and click the **Amazon EC2** tab.
2. Click **Instances** in the **Navigation** pane.

   The console displays a list of running instances.
3. Select the instance to terminate and click **Terminate**.

   Amazon EC2 begins terminating the instance. After termination is complete, the instance status changes to `terminated`.

You're Finished!

Congratulations! You successfully launched, accessed, and terminated an instance. For information on major Amazon EC2 features that were not covered in this guide and how to continue, see Where Do I Go from Here? (p. 45). If you have not already done so, we recommend installing the command line tools. For more information, see Setting up the Tools (p. 20)

Please Give Us Your Feedback

Your input is important to us to help make our documentation helpful and easy to use. Please take a minute to give us your feedback on how well we were able to help you get started with Amazon EC2. Just click this Feedback Link link. Thank you.

Getting Started with the Command Line Tools

Topics

This section describes how to get started with the command line tools, a set of tools that you can run from the Linux/UNIX or Windows command line that closely mimic the Amazon EC2 API functions.

This section describes prerequisites, how to set up the tools and their environment, how to run an instance, how to access an instance, and how to terminate it.

Prerequisites

Topics

This document assumes that the reader is comfortable working in a Linux/UNIX or Windows environment.

An installation of a Java 5 compatible Java Runtime Environment (JRE) is required. Additionally, accessing Linux and UNIX instances requires access to an SSH client and accessing Windows instances requires access to a Remote Desktop client. For more information, refer to the two following sections.

As a convention, all command line text is prefixed with a generic **`PROMPT>`** command line prompt. The actual command line prompt on your machine is likely to be different. We also use **`$`** to indicate a Linux/UNIX specific command and **`C:\>`** for a Windows specific command. While we don't currently provide explicit instructions, the tools also work correctly on Mac OS X (which resemble the Linux and UNIX commands). The example output resulting from the command is shown immediately thereafter without any prefix.

Note

If you are using Cygwin, EC2_HOME, EC2_PRIVATE_KEY, and EC2_CERT must use Linux/UNIX paths (e.g,, /usr/bin instead of C:\usr\bin). However, JAVA_HOME should have a Windows path. Additionally, the value of EC2_HOME cannot contain any spaces, even if the value is quoted or the spaces are escaped.

The Java Runtime Environment

The command line tools used in this guide require Java version 5 or later to run. Either a JRE or JDK installation is acceptable. To view and download JREs for a range of platforms, including Linux/UNIX and Windows, go to http://java.sun.com/j2se/1.5.0/.

Setting the Java Home Variable

The command line tools depend on an environment variable (`JAVA_HOME`) to locate the Java runtime. This environment variable should be set to the full path of the directory that contains a sub-directory named `bin` which in turn contains the `java` (on Linux and UNIX) or the `java.exe` (on Windows) executable. You might want to simplify things by adding this directory to your path before other versions of Java.

Following is an example of how to set this environment variable in Linux and UNIX.

```
$ export JAVA_HOME=<PATH>
```

Following is an example of the syntax in Windows.

```
C:\> set JAVA_HOME=<PATH>
```

You can confirm this by running **`$JAVA_HOME/bin/java -version`** and checking the output.

```
$ $JAVA_HOME/bin/java -version
java version "1.5.0_09"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_09-b03)
Java HotSpot(TM) Client VM (build 1.5.0_09-b03, mixed mode, sharing)
```

The syntax is different on Windows, but the output is similar.

```
C:\> %JAVA_HOME%\bin\java -version
java version "1.5.0_09"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_09-b03)
Java HotSpot(TM) Client VM (build 1.5.0_09-b03, mixed mode, sharing)
```

Accessing Linux and UNIX Instances through SSH Clients

For some of the examples illustrated in this guide you'll need access to an SSH client. Most Linux and UNIX installations include an SSH client by default. If yours does not, the OpenSSH project provides a free implementation of the full suite of SSH tools. For more information, go to the http://www.openssh.org.

Windows users can download and install PuTTY, a free SSH client. To download the client and installation instructions, go to the http://www.chiark.greenend.org.uk/~sgtatham/putty/. For information on how to use PuTTY with Amazon EC2, see Appendix: PuTTY (p. 47).

Accessing Windows Instances through Remote Desktop Clients

Some of the examples in this guide require a Remote Desktop client. If you do not have one, go to the Microsoft home page.

Setting up the Tools

Topics

One step remains before you'll be able to use Amazon EC2. You need to get our command line tools and set them up to use your AWS account.

How to Get the Command Line Tools

The command line tools are available as a ZIP file in the Amazon EC2 Resource Center. These tools are written in Java and include shell scripts for both Windows 2000/XP and Linux/UNIX/Mac OSX. The ZIP file is self-contained; no installation is required. You just download it and unzip it.

Some additional setup is required in order for the tools to use your AWS account credentials. These are discussed next.

How to Tell the Tools Where They Live

The command line tools depend on an environment variable (`EC2_HOME`) to locate supporting libraries. You'll need to set this environment variable before you can use the tools. This should be set to the path of the directory into which the command line tools were unzipped. This directory is named `ec2-api-tools-A.B-nnnn` (`A`, `B` and `n` are version/release numbers), and contains sub-directories named `bin` and `lib`.

On Linux and UNIX, you can set this environment variable as follows.

```
$ export EC2_HOME=<path-to-tools>
```

On Windows the syntax is slightly different.

```
C:\> set EC2_HOME=<path-to-tools>
```

In addition, to make your life a little easier, you probably want to add the tools' `bin` directory to your system `PATH`. The rest of this guide assumes is done.

On Linux and UNIX, you can update your PATH as follows.

```
$ export PATH=$PATH:$EC2_HOME/bin
```

On Windows the syntax is slightly different.

```
C:\> set PATH=%PATH%;%EC2_HOME%\bin
```

Note

The Windows environment variables are reset when you close the command window. You might want to set them permanently.

How to Tell the Tools Who You Are

The command line tools need access to the private key and X.509 certificate you generated after signing up for the Amazon EC2 service (see Setting up an Account (p. 8)).

Since there's nothing stopping you from having more than one AWS account, you need to identify yourself to the command line API tools so they know which credentials to use for requests. It's possible, but tedious, to provide this information on the command line every time you invoke the tools. But it's far simpler to set up some environment variables and be done with it.

Two environment variables are supported to make this possible. They can be set to point at your private key and certificate. If these environment variables are set, the tools use their values to find the relevant credentials. The environment variable EC2_PRIVATE_KEY should reference your private key file, and EC2_CERT should reference your X.509 certificate.

On Linux and UNIX, you can set these environment variables as follows.

```
$ export EC2_PRIVATE_KEY=~/.ec2/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem
$ export EC2_CERT=~/.ec2/cert-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem
```

On Windows the syntax is slightly different.

```
C:\> set EC2_PRIVATE_KEY=c:\ec2\pk-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem
C:\> set EC2_CERT=c:\ec2\cert-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem
```

How to Change the Region

By default, the Amazon EC2 tools use the Eastern United States region (us-east-1) with the us-east-1.ec2.amazonaws.com service endpoint URL. This section describes how to specify a different region by changing the service endpoint URL.

To specify a different region

1. View available regions by entering the following:

   ```
   PROMPT> ec2-describe-regions

   REGION        us-east-1        us-east-1.ec2.amazonaws.com
   REGION        eu-west-1        eu-west-1.ec2.amazonaws.com
   ```

2. If you want to change the service endpoint URL on Linux and UNIX, set the EC2_URL environment variable as follows:

   ```
   $ export EC2_URL=https://<service_endpoint>
   ```

3. If you want to change the service endpoint URL on Windows, set the EC2_URL environment variable as follows:

```
C:\> set EC2_URL=https://<service_endpoint>
```

You're ready to start using Amazon EC2.

Running an Instance

Topics

- Linux and UNIX (p. 22)
- Windows (p. 34)

The following sections provided guided examples of how to launch both Linux/UNIX and Windows instances.

Linux and UNIX

Topics

- Running a Linux/UNIX Instance (p. 22)
- Bundling an AMI (p. 29)

This section describes how to launch and access Linux and UNIX instances using the command line tools. For information on how to launch an instance using the AWS Management Console, see Getting Started with the Console (p. 11).

Running a Linux/UNIX Instance

Topics

- Before We Begin (p. 22)
- How to Find a Suitable AMI (p. 23)
- How to Generate an SSH Key Pair (p. 24)
- How to Run an Instance (p. 26)
- How to Authorize Network Access to Your Instances (p. 27)
- How to Connect to your Instance (p. 27)
- Loading Software and Making Changes (p. 28)
- What's Next (p. 29)

This section describes how to run an instance that uses Linux or UNIX.

Before We Begin

Before running an instance, verify the requirements in the following table.

Verification Process

1	Ensure you have a version 1.5.0 compatible Java Runtime installation, and that the JAVA_HOME environment variable has been correctly set. If not, see The Java Runtime Environment (p. 19).

2	Ensure you have an active Amazon Web Services Account, and that you've signed up for both Amazon S3 and Amazon EC2. If not, see Setting up an Account (p. 8).
3	Ensure that you have created a directory called `.ec2` in your home directory for Linux/UNIX or a directory without spaces in Windows (e.g., C:\EC2), that contains your X.509 certificate and private key, and that they're named correctly. If not, see Prerequisites (p. 18).
4	Ensure that the `EC2_HOME` environment variable are correctly set. If not, see How to Tell the Tools Where They Live (p. 20).
5	Ensure that the `EC2_CERT` and `EC2_PRIVATE_KEY` environment variables are correctly set. If not, see How to Tell the Tools Who You Are (p. 21).

Once these are correct, you are ready to launch your first instance.

How to Find a Suitable AMI

This section describes how to find a baseline AMI that is suitable for your requirements.

To find a suitable AMI

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

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

IMAGE    ami-2c5fba45    ec2-public-images/demo-paid-AMI-v1.07.manifest.xml
    amazon    available    public    A79EC0DB    i386    machine
IMAGE    ami-bd9d78d4    ec2-public-images/demo-paid-AMI.manifest.xml
 amazon    available    public    A79EC0DB    i386    machine
IMAGE    ami-2f5fba46    ec2-public-images/developer-image-i386-
v1.07.manifest.xml    amazon     available    public        i386    machine
IMAGE    ami-26b6534f    ec2-public-images/developer-image.manifest.xml
 amazon    available     public        i386    machine
IMAGE    ami-f51aff9c    ec2-public-images/fedora-8-i386-base-
v1.06.manifest.xml    amazon    available    public        i386    machine
   aki-a71cf9ce    ari-a51cf9cc
IMAGE    ami-2b5fba42    ec2-public-images/fedora-8-i386-base-
v1.07.manifest.xml    amazon     available    public        i386    machine
    aki-a71cf9ce    ari-a51cf9cc
IMAGE    ami-f21aff9b    ec2-public-images/fedora-8-x86_64-base-
v1.06.manifest.xml    amazon     available    public        x86_64
 machine    aki-b51cf9dcari-b31cf9da
IMAGE    ami-2a5fba43    ec2-public-images/fedora-8-x86_64-base-
v1.07.manifest.xml    amazon    available    public        x86_64
 machine    aki-b51cf9dcari-b31cf9da
IMAGE    ami-a21affcb    ec2-public-images/fedora-core-6-x86_64-base-
v1.06.manifest.xml    amazon    available    public        x86_64
 machine    aki-a53adfccari-a23adfcb
IMAGE    ami-2d5fba44    ec2-public-images/fedora-core-6-x86_64-base-
v1.07.manifest.xml     amazon    available    public        x86_64
 machine    aki-a53adfccari-a23adfcb
IMAGE    ami-225fba4b    ec2-public-images/fedora-core4-apache-mysql-
v1.07.manifest.xml    amazon    available    public        i386    machine
IMAGE    ami-25b6534c    ec2-public-images/fedora-core4-apache-
mysql.manifest.xml    amazon    available    public        i386    machine
IMAGE    ami-2e5fba47    ec2-public-images/fedora-core4-apache-
v1.07.manifest.xml    amazon    available    public        i386    machine
```

```
IMAGE    ami-23b6534a    ec2-public-images/fedora-core4-apache.manifest.xml
    amazon    available    public        i386    machine
IMAGE    ami-215fba48    ec2-public-images/fedora-core4-base-
v1.07.manifest.xml    amazon    available    public        i386    machine
IMAGE    ami-20b65349    ec2-public-images/fedora-core4-base.manifest.xml
  amazon    available    public        i386    machine
IMAGE    ami-205fba49    ec2-public-images/fedora-core4-i386-base-
v1.07.manifest.xml    amazon    available    public        i386    machine
   aki-9b00e5f2
IMAGE    ami-255fba4c    ec2-public-images/fedora-core4-mysql-
v1.07.manifest.xml    amazon    available    public        i386    machine
IMAGE    ami-22b6534b    ec2-public-images/fedora-core4-mysql.manifest.xml
   amazon    available    public        i386    machine
IMAGE    ami-36ff1a5f    ec2-public-images/fedora-core6-base-
x86_64.manifest.xml    amazon    available    public        x86_64
 machine
IMAGE    ami-235fba4a    ec2-public-images/getting-started-
v1.07.manifest.xml    amazon    available    public        i386    machine
IMAGE    ami-2bb65342    ec2-public-images/getting-started.manifest.xml
 amazon    available    public        i386    machine
```

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

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

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

How to Generate an SSH Key Pair

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

To generate a key pair using `gsg-keypair`

1. Enter the following information.

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

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

```
KEYPAIR gsg-keypair
 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f
-----BEGIN RSA PRIVATE KEY-----
MIIEoQIBAAKCAQBuLFg5ujHrtm1jnutSuoO8Xe56LlT+HM8v/xkaa39EstM3/
aFxTHgElQiJLChp
HungXQ29VTc8rc1bW0lkdi23OH5eqkMHGhvEwqa0HWASUMll4o3o/IX+0f2UcPoKCOVUR
+jx71Sg
5AU52EQfanIn3ZQ8lFW7Edp5a3q4DhjGlUKToHVbicL5E+g45zfB95wIyywWZfeW/
UUF3LpGZyq/
ebIUlq1qTbHkLbCC2r7RTn8vpQWp47BGVYGtGSBMpTRP5hnbzzuqj3itkiLHjU39S2sJCJ0TrJx5
i8BygR4s3mHKBj8l
+ePQxG1kGbF6R4yg6sECmXn17MRQVXODNHZbAgMBAAECggEAY1tsiUsIwDl5
91CXirkYGuVfLyLflXenxfI50mDFms/mumTqloHO7tr0oriHDR5K7wMcY/
YY5YkcXNo7mvUVD1pM
```

```
ZNUJs7rw9gZRTrf7LylaJ58kOcyajw8TsC4e4LPbFaHwS1d6K8rXh64o6WgW4SrsB6ICmr1kGQI7
3wcfgt5ecIu4TZf0OE9IHjn+2eRlsrjBdeORi7KiUNC/pAG23I6MdDOFEQRcCSigCj+4/
mciFUSA
SWS4dMbrpb9FNSIcf9dcLxVM7/6KxgJNfZc9XWzUw77Jg8x92Zd0fVhHOux5IZC
+UvSKWB4dyfcI
tE8C3p9bbU9VGyY5vLCAiIb4qQKBgQDLiO24GXrIkswF32YtBBMuVgLGCwU9h9HlO9mKAc2m8Cm1
jUE5IpzRjTedc9I2qiIMUTwtgnw42auSCzbUeYMURPtDqyQ7p6AjMujp9EPemcSVOK9vXYL0Ptco
xW9MC0dtV6iPkCN7gOqiZXPRKaFbWADp16p8UAIvS/
a5XXk5jwKBgQCKkpHi2EISh1uRkhxljyWC
iDCiK6JBRsMvpLbc0v5dKwP5alo1fmdR5PJaV2qvZSj5CYNpMAy1/EDNTY5OSIJU
+0KFmQbyhsbm
rdLNLDL4+TcnT7c62/aH01ohYaf/VCbRhtLlBfqGoQc7+sAc8vmKkesnF7CqCEKDyF/
dhrxYdQKB
gC0iZzzNAapayz1+JcVTwwEid6j9JqNXbBc+Z2YwMi+T0Fv/P/hwkX/ypeOXnIUcw0Ih/
YtGBVAC
DQbsz7LcY1HqXiHKYNWNvXgwwO
+oiChjxvEkSdsTTIfnK4VSCvU9BxDbQHjdiNDJbL6oar92UN7V
rBYvChJZF7LvUH4YmVpHAoGAbZ2X7XvoeEO+uZ58/
BGKOIGHByHBDiXtzMhdJr15HTYjxK7OgTZm
gK+8zp4L9IbvLGDMJO8vft32XPEWuvI8twCzFH+CsWLQADZMZKSsBasOZ/h1FwhdMgCMcY
+Qlzd4
JZKjTSu3i7vhvx6RzdSedXEMNTZWN4qlIx3kR5aHcukCgYA9T
+Zrvm1F0seQPbLknn7EqhXIjBaT
P8TTvW/6bdPi23ExzxZn7KOdrfclYRph1LHMpAONv/x2xALIf91UB
+v5ohy1oDoasL0gij1houRe
2ERKKdwz0ZL9SWq6VTdhr/5G994CK72fy5WhyERbDjUIdHaK3M849JJuf8cSrvSb4g==
-----END RSA PRIVATE KEY-----
```

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

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

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

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

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

   ```
    -----BEGIN RSA PRIVATE KEY-----
   MIIEoQIBAAKCAQBuLFg5ujHrtm1jnutSuoO8Xe56LlT+HM8v/xkaa39EstM3/
   aFxTHgElQiJLChp
   HungXQ29VTc8rc1bW0lkdi23OH5eqkMHGhvEwqa0HWASUMll4o3o/IX+0f2UcPoKCOVUR
   +jx71Sg
   5AU52EQfanIn3ZQ8lFW7Edp5a3q4DhjGlUKToHVbicL5E+g45zfB95wIyywWZfeW/
   UUF3LpGZyq/
   ebIUlq1qTbHkLbCC2r7RTn8vpQWp47BGVYGtGSBMpTRP5hnbzzuqj3itkiLHjU39S2sJCJ0TrJx5
   i8BygR4s3mHKBj8l
   +ePQxG1kGbF6R4yg6sECmXn17MRQVXODNHZbAgMBAAECggEAY1tsiUsIwDl5
   91CXirkYGuVfLyLflXenxfI50mDFms/mumTqloHO7tr0oriHDR5K7wMcY/
   YY5YkcXNo7mvUVD1pM
   ZNUJs7rw9gZRTrf7LylaJ58kOcyajw8TsC4e4LPbFaHwS1d6K8rXh64o6WgW4SrsB6ICmr1kGQI7
   3wcfgt5ecIu4TZf0OE9IHjn+2eRlsrjBdeORi7KiUNC/pAG23I6MdDOFEQRcCSigCj+4/
   mciFUSA
   SWS4dMbrpb9FNSIcf9dcLxVM7/6KxgJNfZc9XWzUw77Jg8x92Zd0fVhHOux5IZC
   +UvSKWB4dyfcI
   tE8C3p9bbU9VGyY5vLCAiIb4qQKBgQDLiO24GXrIkswF32YtBBMuVgLGCwU9h9HlO9mKAc2m8Cm1
   jUE5IpzRjTedc9I2qiIMUTwtgnw42auSCzbUeYMURPtDqyQ7p6AjMujp9EPemcSVOK9vXYL0Ptco
   ```

```
xW9MC0dtV6iPkCN7gOqiZXPRKaFbWADp16p8UAIvS/
a5XXk5jwKBgQCKkpHi2EISh1uRkhxljyWC
iDCiK6JBRsMvpLbc0v5dKwP5alo1fmdR5PJaV2qvZSj5CYNpMAy1/EDNTY5OSIJU
+0KFmQbyhsbm
rdLNLDL4+TcnT7c62/aH01ohYaf/VCbRhtLlBfqGoQc7+sAc8vmKkesnF7CqCEKDyF/
dhrxYdQKB
gC0iZzzNAapayz1+JcVTwwEid6j9JqNXbBc+Z2YwMi+T0Fv/P/hwkX/ypeOXnIUcw0Ih/
YtGBVAC
DQbsz7LcY1HqXiHKYNWNvXgwwO
+oiChjxvEkSdsTTIfnK4VSCvU9BxDbQHjdiNDJbL6oar92UN7V
rBYvChJZF7LvUH4YmVpHAoGAbZ2X7XvoeEO+uZ58/
BGKOIGHByHBDiXtzMhdJr15HTYjxK7OgTZm
gK+8zp4L9IbvLGDMJO8vft32XPEWuvI8twCzFH+CsWLQADZMZKSsBasOZ/h1FwhdMgCMcY
+Qlzd4
JZKjTSu3i7vhvx6RzdSedXEMNTZWN4qlIx3kR5aHcukCgYA9T
+Zrvm1F0seQPbLknn7EqhXIjBaT
P8TTvW/6bdPi23ExzxZn7KOdrfclYRph1LHMpAONv/x2xALIf91UB
+v5ohy1oDoasL0gij1houRe
2ERKKdwz0ZL9SWq6VTdhr/5G994CK72fy5WhyERbDjUIdHaK3M849JJuf8cSrvSb4g==
-----END RSA PRIVATE KEY-----
```

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

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

   ```
   $  chmod 600 id_rsa-gsg-keypair ; ls -l id_rsa-gsg-keypair
   ```

   You receive output similar to the following example.

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

Note

If you are using PuTTY in Windows, convert the private key to PuTTY's format. For more information on using PuTTy with Amazon EC2, see Appendix: PuTTY (p. 47) .

How to Run an Instance

This section describes how to run an instance that uses Linux or UNIX.

To launch an instance of your AMI

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

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

   Amazon EC2 returns output similar to the following example.

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

2. Look for the instance ID in the second field and write it down.

   You use it to manipulate this instance (including terminating it when you are finished).

   It takes a few minutes for the instance to launch.

3. The following command displays the launch status of the instance.

   ```
   PROMPT>  ec2-describe-instances i-ae0bf0c7
   RESERVATION     r-7430c31d      924417782495    default
   ```

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

Important

After launching an instance, you are billed hourly for running time. If you leave this tutorial at any time, make sure you terminate any instances you have started as described in How to Terminate Your Instances (p. 43).

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

How to Authorize Network Access to Your Instances

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

To authorize access to your instance

- Enter the `ec2-authorize` command.

```
PROMPT>  ec2-authorize default -p 22
PERMISSION     default  ALLOWS  tcp     22      22      FROM    CIDR
 0.0.0.0/0
PROMPT>  ec2-authorize default -p 80
PERMISSION     default  ALLOWS  tcp     80      80      FROM    CIDR
 0.0.0.0/0
```

Since we didn't specify otherwise, your instance was launched in your `default` group. The first command authorizes network access to instances in your default group on the standard SSH port (22). Similarly, the second command opens up the standard HTTP port (80). For more details on controlling network security groups, go to the Amazon Elastic Compute Cloud User Guide or Amazon Elastic Compute Cloud Developer Guide.

Caution

In this example, you enable any IP address to access port 22 of the instance. Although this might be acceptable for testing purposes, it is extremely unsafe for production environments. For production systems, you must obtain your public IP address ranges and grant access to those ranges only. For example, if your IP address is 123.123.123.123, you specify `123.123.123.123/32`.

How to Connect to your Instance

Now that the instance is running and access is authorized, you are ready to connect.

To connect to your instance

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

A webpage welcoming you to your instance displays.

Note

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

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

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

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

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

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

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

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

Note

Your machine might have a different name for the preceding **ssh** command or even use different command line options. Consult the documentation for your machine or download one of the clients described in Accessing Linux and UNIX Instances through SSH Clients (p. 19) if you are unsure whether you have such a client installed. For more information about using PuTTY on Windows, see Appendix: PuTTY (p. 47).

Loading Software and Making Changes

Now that you are logged into the instance, you can load software and make changes as you would with any server. When your changes are finished, you can bundle them as a new AMI and launch an identical copy at any time.

What's Next

You've set up the tools and used them to run an instance based on a public AMI. You have learned enough to successfully use Amazon EC2 to run as many standard Linux/UNIX instances as you wish. You can run instances based on any of the public AMIs by following this process.

The next section builds on this success by having you connect to the running instance and customize it to create your own image. If you want to skip creating an AMI, make sure to terminate any instances that you started. For more information, see Cleaning Up (p. 43).

Bundling an AMI

Topics

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

Bunndling involves creating a new image or modifying an existing one, preparing for bundling, bundling the image, uploading the AMI to Amazon S3, and registering the new AMI.

Note

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

If you want to skip creating an AMI, make sure to terminate any instances that you started. For more information, see Cleaning Up (p. 43).

Modifying an Existing Image

This section describes how to modfy an existing AMI which can be used as a baseline for your new AMI.

To modify an image

1. Select an image to serve as the basis for the new one.

   The image used to create this instance contains a default web site which you are going to modify.
2. Log in to your instance as described in How to Connect to your Instance (p. 27).
3. Modify the main web page by replacing some of the static content with your name to personalize it. Use the following command.

   Don't worry too much about what exactly it does, but remember to replace the text ***<YourName>*** with your own name.

   ```
   # sed -i -e 's/Congratulations!/Congratulations <YourName>!/' /var/www/
   html/index.html
   ```
4. Check that the file is updated by confirming that the date and time displayed in the following output example matches the current date.

   ```
   # ls -l /var/www/html/index.html
   ```

```
  -rw-rw-r--  1 root root       1872 Jun 21 09:33 /var/www/html/index.html
  # date
  Wed Jun 21 09:33:42 EDT 2006
```

Preparing for Bundling

Once your system has been suitably modified, a system snapshot needs to be created and packaged into an AMI by using the **ec2-bundle-vol** utility.

ec2-bundle-vol encrypts and signs the image to ensure it cannot be tampered with and that only you and Amazon EC2 can decrypt it.

Let's assume the private key and X.509 certificate are contained in the files `pk-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem` and `cert-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem`. Copy your private key and certificate to the machine being bundled.

```
PROMPT> scp -i id_rsa-gsg-keypair pk-
HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem cert-
HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem
 root@domU-12-34-31-00-00-05.compute-1.amazonaws.com:/mnt
pk-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem                         100%  717
 0.7KB/s   00:00
cert-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem                       100%  684
 0.7KB/s   00:00
```

Note

Your machine may have a different name for the preceding **scp** command or even use different command line options. Consult the documentation for your machine or download one of the clients described in Accessing Linux and UNIX Instances through SSH Clients (p. 19). If you are running Windows, see Appendix: PuTTY (p. 47) for help on using PuTTY and pscp (PuTTY's secure copy tool).

Note

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

Bundling

At this point, the machine image has been modified and the private key and X.509 certificate uploaded. The AMI can now be bundled, using your *AWS account ID* as your username (*not* your AWS Access Key ID).

```
# ec2-bundle-vol -d /mnt -k /mnt/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem -c /
mnt/cert-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem -u AIDADH4IGTRXXKCD -r i386 -p
 sampleimage
Copying / into the image file /mnt/sampleimage.img...
Excluding:
     /sys
     /dev/shm
     /proc
     /dev/pts
     /proc/sys/fs/binfmt_misc
     /dev
     /media
     /mnt
     /proc
     /sys
     /mnt/sampleimage.img
     /mnt/img-mnt
1+0 records in
1+0 records out
mke2fs 1.38 (30-Jun-2005)
warning: 256 blocks unused.

Splitting /mnt/sampleimage.gz.crypt...
Created sampleimage.part.00
Created sampleimage.part.01
Created sampleimage.part.02
Created sampleimage.part.03
...
Created sampleimage.part.22
Created sampleimage.part.23
Generating digests for each part...
Digests generated.
Creating bundle manifest...
Bundle Volume complete.
```

This command may take several minutes to complete. Once the bundling process is complete, the AMI, its associated manifest file, and image parts are in the `/mnt` directory. You can confirm this using the command in the following example.

```
# ls -l /mnt/sampleimage.*
-rw-r--r--  1 root root 1611661312 Jun 20 10:12 /mnt/sampleimage.image
-rw-r--r--  1 root root   10485760 Jun 20 10:12 /mnt/sampleimage.part.00
-rw-r--r--  1 root root   10485760 Jun 20 10:12 /mnt/sampleimage.part.01
-rw-r--r--  1 root root   10485760 Jun 20 10:12 /mnt/sampleimage.part.02
-rw-r--r--  1 root root   10485760 Jun 20 10:12 /mnt/sampleimage.part.03
...
-rw-r--r--  1 root root   10485760 Jun 20 10:12 /mnt/sampleimage.part.22
-rw-r--r--  1 root root   10485760 Jun 20 10:12 /mnt/sampleimage.part.23
-rw-r--r--  1 root root       2970 Jun 20 10:12 /mnt/sampleimage.manifest
```

Uploading the AMI to Amazon S3

All AMIs are loaded from Amazon S3 storage. You need to upload the newly bundled AMI to an existing account on Amazon S3 such as the one you created in Signing up for Amazon S3 (p. 9).

Amazon S3 stores data objects in buckets, which are similar in concept to directories. You'll need to specify a bucket name in the following example as <your-s3-bucket>. Buckets have globally unique names and are owned by unique users. If you have used S3 before, you can use any of your existing buckets. The **ec2-upload-bundle** utility uploads the bundled AMI to a specified bucket. If the specified bucket does not exist, it creates it. If the specified bucket belongs to another user **ec2-upload-bundle** will fail, and must use a different name.

For this, you'll need your AWS Access Key ID (<aws-access-key-id>) and AWS Secret Access Key (<aws-secret-access-key>). For information on how to find these keys, see Signing up for Amazon S3 (p. 9).

The upload process can be quite lengthy, but Amazon EC2 provides continuous feedback until the upload completes as shown in the following example.

```
# ec2-upload-bundle -b <your-s3-bucket> -m /mnt/sampleimage.manifest.xml -
a <aws-access-key-id> -s <aws-secret-access-key>
Encrypting bundle manifest...
```

```
Completed encryption.
Uploading encrypted manifest...
Uploaded encrypted manifest to https://s3.amazonaws.com/<your-s3-bucket>/mnt/
sampleimage.manifest.xml
Uploading bundled AMI parts to https://s3.amazonaws.com/<your-s3-bucket>/
sampleimage...
Uploaded sampleimage.part.00 to https://s3.amazonaws.com/<your-s3-bucket>/
sampleimage.part.00.
Uploaded sampleimage.part.01 to https://s3.amazonaws.com/<your-s3-bucket>/
sampleimage.part.01.
Uploaded sampleimage.part.02 to https://s3.amazonaws.com/<your-s3-bucket>/
sampleimage.part.02.
Uploaded sampleimage.part.03 to https://s3.amazonaws.com/<your-s3-bucket>/
sampleimage.part.03.
...
Uploaded sampleimage.part.23 to https://s3.amazonaws.com/<your-s3-bucket>/
sampleimage.part.23.
Uploaded sampleimage.part.24 to https://s3.amazonaws.com/<your-s3-bucket>/
sampleimage.part.24.
Upload Bundle complete.
```

Note

Bear in mind that once the bundle has been uploaded to Amazon S3, you will be charged for the storage. You may remove the image from Amazon S3 as described in How to Remove Your AMI from Amazon S3 (p. 44).

At this point your new image is ready to be registered and launched, and you can log out of your ssh session.

Amazon S3 Bucket Naming Restrictions

To comply with Amazon S3 requirements, bucket names must:

- Contain lowercase letters, numbers, periods (.), underscores (_), and dashes (-)
- Start with a number or letter
- Be between 3 and 255 characters long
- Not be in an IP address style (e.g., "192.168.5.4")

To conform with DNS requirements, we recommend following these additional guidelines when creating buckets:

- Bucket names should not contain underscores (_)
- Bucket names should be between 3 and 63 characters long
- Bucket names should not end with a dash
- Bucket names cannot contain dashes next to periods (e.g., "my-.bucket.com" and "my.-bucket" are invalid)

Registering the AMI

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

Note

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

To register the AMI you uploaded to Amazon S3

- Select the AMI in the AWS Management Console and click **Register New AMI** or execute the following command:

  ```
  PROMPT> ec2-register <your-s3-bucket>/sampleimage.manifest.xml
  IMAGE ami-2bb65342
  ```

  Amazon EC2 returns an AMI identifier (the value next to the `IMAGE` tag on the command line) that can be used to run instances.

Running Instances

You can now run an instance of the modified AMI by specifying the image identifier you received when you registered the image.

To run an instance

- Execute the following command, substituting the image identifier with that received in Registering the AMI (p. 32):

  ```
  PROMPT> ec2-run-instances ami-5bae4b32
  RESERVATION     r-3d30c354      924417782495    default
  INSTANCE i-10a64379 ami-5bae4b32 pending 0 m1.small 2007-07-11T16:40:44+0000
   us-east-1a
  ```

  Amazon EC2 starts a single instance based on your newly created AMI and provides you with an instance identifier, the value immediately to the right of the `INSTANCE` tag, that can be used to monitor the status of the running instance. For information on confirming the instance is up and running, see How to Run an Instance (p. 26),

You now have your very own image starting up and getting ready. You can monitor its status until it's ready and then connect to it with your web browser to confirm the changes you had made are actually live. If you want to ssh in and take a look around, you can do so using the key pair you created in the How to Connect to your Instance (p. 27). The keypair you launched that instance with was included in your new AMI during the bundling process.

Congratulations

You have successfully built and deployed your very own AMI, and run instances based on it. This custom AMI is private to your account. You can build as many custom AMIs as required and use them to launch as many instances as you need.

With these simple building blocks and the other public AMIs made available by Amazon and third parties, you're well positioned to realize the full benefits of Amazon EC2.

Note

AMIs published by third parties should be launched with caution. Amazon does not vet public AMIs. We recommend checking the forums for community feedback on a public AMI before launching it, and taking necessary precautions after launching it.

For brief information about major Amazon EC2 features that you might want to use, see Where Do I Go from Here? (p. 45). For information about Amazon EC2 command line tools, go to the Amazon Elastic Compute Cloud Command Line Reference. For more information about APIs, go to the Amazon Elastic Compute Cloud Developer Guide and Amazon Elastic Compute Cloud API Reference. For general information, go to the Amazon Elastic Compute Cloud User Guide.

Note

Make sure to terminate any instances that you started because you are billed for their usage. For more information, see Cleaning Up (p. 43).

Windows

This section describes how to launch and access Windows instances using the command line tools. For information on how to launch an instance using the AWS Management Console, see Getting Started with the Console (p. 11).

Note

This guide describes how to launch and access Windows instances using the Amazon EC2 command line tools. To launch Windows instances using a GUI, we recommend downloading the latest Elasticfox Firefox extension.

Running a Windows Instance

Topics

This section describes how to run an instance that uses Windows.

Before We Begin

Before running an instance, verify the requirements in the following table.

Verification Process

1	Ensure you have a version 1.5.0 compatible Java Runtime installation, and that the `JAVA_HOME` environment variable has been correctly set. If not, see The Java Runtime Environment (p. 19).
2	Ensure you have an active Amazon Web Services Account, and that you've signed up for both Amazon S3 and Amazon EC2. If not, see Setting up an Account (p. 8).
3	Ensure that you have created a directory called `.ec2` in your home directory for Linux/UNIX or a directory without spaces in Windows (e.g., C:\EC2), that contains your X.509 certificate and private key, and that they're named correctly. If not, see Prerequisites (p. 18).
4	Ensure that the `EC2_HOME` environment variable has been correctly set. If not, see How to Tell the Tools Where They Live (p. 20).

5	Ensure that the `EC2_CERT` and `EC2_PRIVATE_KEY` environment variables have been correctly set. If not, see How to Tell the Tools Who You Are (p. 21).

Once these are correct, you are ready to launch your first instance.

How to Find a Suitable AMI

This section describes how to find an AMI to use for this exercise.

To find a suitable AMI

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

   ```
   C:\> ec2-describe-images -o self -o amazon | findstr /i windows
   
   IMAGE   ami-e3698d8a    ec2-public-windows-images/Server2003r2-i386-Win-
   v1.02.manifest.xml       amazon  available       public          i386
    machine     windows
   IMAGE   ami-e5698d8c    ec2-public-windows-images/Server2003r2-i386-
   WinAuth-v1.02.manifest.xml   amazon  available       public          i386
     machine     windows
   IMAGE   ami-ed698d84    ec2-public-windows-images/Server2003r2-x86_64-Win-
   v1.02.manifest.xml     amazon  available       public          x86_64
    machine     windows
   IMAGE   ami-ec698d85    ec2-public-windows-images/Server2003r2-x86_64-
   WinAuth-v1.02.manifest.xml        amazon  available       public
    x86_64  machine                      windows
   IMAGE   ami-e4698d8d    ec2-public-windows-images/SqlSvrExp2003r2-i386-
   Win-v1.02.manifest.xml    amazon  available       public          i386
    machine     windows
   ```

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

2. Look for the 32-bit Windows Anonymous AMI and note the corresponding value in the second column.

   This is the AMI ID you need. In this example, it is `ami-e3698d8a`.

How to Generate an SSH Key Pair

Amazon EC2 uses the SSH key pair to enable you to securely get your Windows password. Every key pair you generate requires a name. Be sure to choose a name that is easy to remember.

To generate a key pair using `gsg-keypair`

1. Enter the following information.

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

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

   ```
   KEYPAIR gsg-keypair
    1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f
   -----BEGIN RSA PRIVATE KEY-----
   MIIEoQIBAAKCAQBuLFg5ujHrtm1jnutSuoO8Xe56LlT+HM8v/xkaa39EstM3/
   aFxTHgElQiJLChp
   HungXQ29VTc8rc1bW0lkdi23OH5eqkMHGhvEwqa0HWASUMll4o3o/IX+0f2UcPoKCOVUR
   +jx71Sg
   ```

```
5AU52EQfanIn3ZQ8lFW7Edp5a3q4DhjGlUKToHVbicL5E+g45zfB95wIyywWZfeW/
UUF3LpGZyq/
ebIUlq1qTbHkLbCC2r7RTn8vpQWp47BGVYGtGSBMpTRP5hnbzzuqj3itkiLHjU39S2sJCJ0TrJx5
i8BygR4s3mHKBj8l
+ePQxG1kGbF6R4yg6sECmXn17MRQVXODNHZbAgMBAAECggEAY1tsiUsIwDl5
91CXirkYGuVfLyLflXenxfI50mDFms/mumTqloHO7tr0oriHDR5K7wMcY/
YY5YkcXNo7mvUVD1pM
ZNUJs7rw9gZRTrf7LylaJ58kOcyajw8TsC4e4LPbFaHwS1d6K8rXh64o6WgW4SrsB6ICmr1kGQI7
3wcfgt5ecIu4TZf0OE9IHjn+2eRlsrjBdeORi7KiUNC/pAG23I6MdDOFEQRcCSigCj+4/
mciFUSA
SWS4dMbrpb9FNSIcf9dcLxVM7/6KxgJNfZc9XWzUw77Jg8x92Zd0fVhHOux5IZC
+UvSKWB4dyfcI
tE8C3p9bbU9VGyY5vLCAiIb4qQKBgQDLiO24GXrIkswF32YtBBMuVgLGCwU9h9HlO9mKAc2m8Cm1
jUE5IpzRjTedc9I2qiIMUTwtgnw42auSCzbUeYMURPtDqyQ7p6AjMujp9EPemcSVOK9vXYL0Ptco
xW9MC0dtV6iPkCN7gOqiZXPRKaFbWADp16p8UAIvS/
a5XXk5jwKBgQCKkpHi2EISh1uRkhxljyWC
iDCiK6JBRsMvpLbc0v5dKwP5alo1fmdR5PJaV2qvZSj5CYNpMAy1/EDNTY5OSIJU
+0KFmQbyhsbm
rdLNLDL4+TcnT7c62/aH01ohYaf/VCbRhtLlBfqGoQc7+sAc8vmKkesnF7CqCEKDyF/
dhrxYdQKB
gC0iZzzNAapayz1+JcVTwwEid6j9JqNXbBc+Z2YwMi+T0Fv/P/hwkX/ypeOXnIUcw0Ih/
YtGBVAC
DQbsz7LcY1HqXiHKYNWNvXgwwO
+oiChjxvEkSdsTTIfnK4VSCvU9BxDbQHjdiNDJbL6oar92UN7V
rBYvChJZF7LvUH4YmVpHAoGAbZ2X7XvoeEO+uZ58/
BGKOIGHByHBDiXtzMhdJr15HTYjxK7OgTZm
gK+8zp4L9IbvLGDMJO8vft32XPEWuvI8twCzFH+CsWLQADZMZKSsBasOZ/h1FwhdMgCMcY
+Qlzd4
JZKjTSu3i7vhvx6RzdSedXEMNTZWN4qlIx3kR5aHcukCgYA9T
+Zrvm1F0seQPbLknn7EqhXIjBaT
P8TTvW/6bdPi23ExzxZn7KOdrfclYRph1LHMpAONv/x2xALIf91UB
+v5ohy1oDoasL0gij1houRe
2ERKKdwz0ZL9SWq6VTdhr/5G994CK72fy5WhyERbDjUIdHaK3M849JJuf8cSrvSb4g==
-----END RSA PRIVATE KEY-----
```

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

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

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

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

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

```
 -----BEGIN RSA PRIVATE KEY-----
MIIEoQIBAAKCAQBuLFg5ujHrtm1jnutSuoO8Xe56LlT+HM8v/xkaa39EstM3/
aFxTHgElQiJLChp
HungXQ29VTc8rc1bW0lkdi23OH5eqkMHGhvEwqa0HWASUMll4o3o/IX+0f2UcPoKCOVUR
+jx71Sg
5AU52EQfanIn3ZQ8lFW7Edp5a3q4DhjGlUKToHVbicL5E+g45zfB95wIyywWZfeW/
UUF3LpGZyq/
ebIUlq1qTbHkLbCC2r7RTn8vpQWp47BGVYGtGSBMpTRP5hnbzzuqj3itkiLHjU39S2sJCJ0TrJx5
i8BygR4s3mHKBj8l
+ePQxG1kGbF6R4yg6sECmXn17MRQVXODNHZbAgMBAAECggEAY1tsiUsIwDl5
91CXirkYGuVfLyLflXenxfI50mDFms/mumTqloHO7tr0oriHDR5K7wMcY/
YY5YkcXNo7mvUVD1pM
ZNUJs7rw9gZRTrf7LylaJ58kOcyajw8TsC4e4LPbFaHwS1d6K8rXh64o6WgW4SrsB6ICmr1kGQI7
```

```
3wcfgt5ecIu4TZf0OE9IHjn+2eRlsrjBdeORi7KiUNC/pAG23I6MdDOFEQRcCSigCj+4/
mciFUSA
SWS4dMbrpb9FNSIcf9dcLxVM7/6KxgJNfZc9XWzUw77Jg8x92Zd0fVhHOux5IZC
+UvSKWB4dyfcI
tE8C3p9bbU9VGyY5vLCAiIb4qQKBgQDLiO24GXrIkswF32YtBBMuVgLGCwU9h9HlO9mKAc2m8Cm1
jUE5IpzRjTedc9I2qiIMUTwtgnw42auSCzbUeYMURPtDqyQ7p6AjMujp9EPemcSVOK9vXYL0Ptco
xW9MC0dtV6iPkCN7gOqiZXPRKaFbWADp16p8UAIvS/
a5XXk5jwKBgQCKkpHi2EISh1uRkhxljyWC
iDCiK6JBRsMvpLbc0v5dKwP5alo1fmdR5PJaV2qvZSj5CYNpMAy1/EDNTY5OSIJU
+0KFmQbyhsbm
rdLNLDL4+TcnT7c62/aH01ohYaf/VCbRhtLlBfqGoQc7+sAc8vmKkesnF7CqCEKDyF/
dhrxYdQKB
gC0iZzzNAapayz1+JcVTwwEid6j9JqNXbBc+Z2YwMi+T0Fv/P/hwkX/ypeOXnIUcw0Ih/
YtGBVAC
DQbsz7LcY1HqXiHKYNWNvXgwwO
+oiChjxvEkSdsTTIfnK4VSCvU9BxDbQHjdiNDJbL6oar92UN7V
rBYvChJZF7LvUH4YmVpHAoGAbZ2X7XvoeEO+uZ58/
BGKOIGHByHBDiXtzMhdJr15HTYjxK7OgTZm
gK+8zp4L9IbvLGDMJO8vft32XPEWuvI8twCzFH+CsWLQADZMZKSsBasOZ/h1FwhdMgCMcY
+Qlzd4
JZKjTSu3i7vhvx6RzdSedXEMNTZWN4qlIx3kR5aHcukCgYA9T
+Zrvm1F0seQPbLknn7EqhXIjBaT
P8TTvW/6bdPi23ExzxZn7KOdrfclYRph1LHMpAONv/x2xALIf91UB
+v5ohy1oDoasL0gij1houRe
2ERKKdwz0ZL9SWq6VTdhr/5G994CK72fy5WhyERbDjUIdHaK3M849JJuf8cSrvSb4g==
-----END RSA PRIVATE KEY-----
```

How to Run the Instance

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

To launch an instance

1. Start the launch by entering the following command:

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

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

   Launching the instance takes a few minutes.

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

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

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

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

Important

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

Example

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

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

The following shows the status of the launch:

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

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

How to Get the Administrator Password

After launching an instance, get its administrator password. This command returns the original password assigned by Amazon EC2.

Note

If you change the administrator password, this command does not retrieve the new password.

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

To get the administrator password

- Enter the following command:

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

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

Amazon EC2 returns the Windows password.

Example

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

```
PROMPT> ec2-get-password -k id_rsa-gsg-keypair i-eb977f82
 Qr89fdS1w
```

How to Authorize Network Access

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

To enable Remote Desktop on port 3389

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

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

Connecting to the Instance

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

To connect to your instance

1. Retrieve the FQDN of your instance.

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

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

   In this example, the FQDN is `ec2-67-202-7-236.compute-1.amazonaws.com`
2. Click **Start**, point to **Programs**, point to **Accessories**, point to **Communications**, and click **Remote Desktop Connection**.

   The **Remote Desktop Connection** dialog box appears.
3. Enter the FQDN in the **Computer** field and click **Connect**.

   The Remote Desktop Connection client connects to the instance.
4. Enter "administrator" as the user name and the instance password. For information about getting the administrator password, see How to Get the Administrator Password (p. 38). .

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

Important

We recommend you exercise extreme care when you change some of the basic Amazon EC2 configuration settings. Otherwise, the AMI might become unbootable or inaccessible from the network once running.

Note

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

Loading Software and Making Changes

Now that you are logged into the Windows instance, you can load software and make changes as you would with any Windows server. When your changes are finished, you can bundle them as a new AMI and launch an identical copy at any time.

Note

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

Congratulations

You've set up the tools and used them to run an instance based on a public AMI. You have learned enough to successfully use Amazon EC2 to run as many Windows instances as you wish. You can run instances based on any of the public AMIs by following this process.

The next section builds on this success by having you bundle the running instance and register it as your own AMI. If you want to stop now, be sure to terminate any instances you have started as described in How to Terminate Your Instances (p. 43).

Bundling an AMI

Topics

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

All AMIs are loaded from Amazon S3 storage. You need to upload the newly bundled AMI to an existing account, such as the one you created in Signing up for Amazon S3 (p. 9).

Amazon S3 stores data objects in buckets, which are similar in concept to directories. You'll need to specify a bucket name in the following example as <your-s3-bucket>. Buckets have globally unique names and are owned by unique users. If you have used S3 before, you can use any of your existing buckets or just give **ec2-bundle-instance** any name that makes sense to you. The **ec2-bundle-instance** utility uploads the bundled AMI to a specified bucket. If the specified bucket does not exist, it creates it. If the specified bucket belongs to another user, the **ec2-bundle-instance** fails and you must use a different name.

For this, you'll need your AWS Access Key ID (<aws-access-key-id>) and AWS Secret Access Key (<aws-secret-access-key>). For information on how to find these keys, see Signing up for Amazon S3 (p. 9).

Caution

Make sure to save your work before bundling. During bundling, Amazon EC2 shuts down the instance and disconnects any remote connection.

Note

During bundling, only the root store (C:\) is bundled. Data on other instance stores is not preserved.

To bundle an AMI

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

> **Note**
>
> We highly recommend changing the password of the AMI. If you choose to use the password that Amazon EC2 provided, write it down so you can access instances launched from this AMI. You cannot get the password of new instances using the `ec2-get-password` command.

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

   > **Note**
   >
   > The `sdelete` utility is available from the sdelete Download Page or the Microsoft Web Site.

3. On the host where you installed the API tools, enter the following command:

   ```
   PROMPT> ec2-bundle-instance <instance_id> -b <bucket_name> -p <bundle_name>
    -o <access_key_id> -w <secret_access_key>
   ```

   The <*instance_id*> is the name of the instance, <*bucket_name*> is the name of the bucket in which to store the AMI, and <*bundle_name*> is the common name for the files to store in Amazon S3.

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

Example

```
PROMPT> ec2-bundle-instance i-eb977f82 -b mybucket -p myimage -
o AKIADQKE4SARGYLE -w eW91dHViZS5jb20vd2F0Y2g/dj1SU3NKMTlzeTNKSQ==
BUNDLE  bun-e3a4418a   i-eb977f82      mybucket myimage
 2008-10-02T09:31:44+0000  2008-10-02T09:31:44+0000        bundling
```

Monitoring a Bundled AMI

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

To view the status

- Enter the following command:

  ```
  # ec2-describe-bundle-tasks
  ```

  Amazon EC2 returns output similar to the following:

  ```
  BUNDLE  bun-55a5403c    i-e3d97c8a      mybucket     winami complete
   2008-08-28T00:59:13+0000        2008-08-28T01:34:30+0000
  ```

Registering the AMI

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

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

To register the AMI

- Execute the following command:

  ```
  PROMPT> ec2-register <your-s3-bucket>/sampleimage.manifest.xml
  IMAGE ami-2bb65342
  ```

  Amazon EC2 returns an AMI identifier, the value next to the `IMAGE` tag (`ami-2bb65342` in the example) that you can use to run instances.

Running Instances

You can now run an instance of the modified AMI by specifying the image identifier you received when you registered the image.

To run an instance

1. Enter the following command:

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

   The <*ami_id*> is the AMI ID of your newly created AMI and <*keypair-name*> is the name of your key pair. The command returns the AMI instance ID, a unique identifier for each launched instance. You use the instance ID to manipulate the instance. This includes viewing the status of the instance, terminating the instance, and so on.

   Launching the instance takes a few minutes.
2. View the progress of the instance by entering the following command:

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

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

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

Important

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

Your instance is starting and getting ready. You can monitor its status until it's ready and then connect to it to confirm the changes you had made are active.

Congratulations

You have successfully built and deployed an AMI, and run instances based on it. This custom AMI is private to your account. You can build as many custom AMIs as required and use them to launch as many instances as you need.

With these simple building blocks and the other public AMIs made available by Amazon and third parties, you're well positioned to realize the full benefits of Amazon EC2.

Note

AMIs published by third parties should be launched with caution. Amazon does not vet public AMIs. We recommend that you check the forums for community feedback on a public AMI before you launch it, and take necessary precautions after you launch it.

For brief information about major Amazon EC2 features that you might want to use, see Where Do I Go from Here? (p. 45). For information about Amazon EC2 command line tools, go to the Amazon Elastic Compute Cloud Command Line Reference. For more information about APIs, go to the Amazon Elastic Compute Cloud Developer Guide and Amazon Elastic Compute Cloud API Reference. For general information, go to the Amazon Elastic Compute Cloud User Guide.

Cleaning Up

Topics

You are charged for running instances and for AMIs that are stored on Amazon S3. This section describes how to terminate your instances, deregister AMIs, and remove AMIs from Amazon S3.

How to Terminate Your Instances

As soon as your instance starts to boot, you are billed for the resources it consumes. Once you have decided that you no longer require its services, you can terminate an instance using the instance identifier you received in How to Run an Instance (p. 26).

Note

You cannot recover a terminated instance. However, you can launch additional instances of an AMI.

To terminate an instance

- Execute the following command:

```
PROMPT>  ec2-terminate-instances i-10a64379
INSTANCE i-10a64379 running shutting-down
```

  It takes a few minutes for the instance to terminate because Amazon EC2 needs to clean-up your data. For information on checking the status of your instance, see How to Run an Instance (p. 26)

Tip

For Linux and UNIX, you can also terminate your instances by logging onto the instances with your ssh tool and running the "shutdown -h" command. Don't forget the "-h", otherwise you put your instance into single user mode, which is quite useless.

```
# /sbin/shutdown -h now
```

For Windows, you can also terminate your instances by selecting **Shutdown** from the **Start** menu or by entering the following from the command line:

```
C:\> shutdown /s /t 0
```

How to Deregister Your AMI

If you no longer need your AMI you should deregister it from Amazon EC2.

For this task you'll need your AWS Access Key ID (<aws-access-key-id>) and AWS Secret Access Key (<aws-secret-access-key>). For more information on how to find these keys, see Signing up for Amazon S3 (p. 9).

To deregister an AMI

- Execute the following command:

```
PROMPT>  ec2-deregister ami-2bb65342
IMAGE ami-2bb65342
```

How to Remove Your AMI from Amazon S3

You can delete your deregistered AMIs from Amazon S3. If you don't, Amazon S3 continues to charge you for the space you are using.

The AMI tools contain a command to delete your image. You can get these from the resource center or they are installed on the instance from which you created the image.

To delete your AMI

- Execute the following command:

```
PROMPT> ec2-delete-bundle -b <your-s3-bucket> -p sampleimage -a <aws-access-
key-id> -s <aws-secret-access-key>
Deleting AMI bundle parts from https://s3.amazonaws.com/<your-s3-bucket>...
Deleted sampleimage.part.00.
Deleted sampleimage.part.01.
Deleted sampleimage.part.02.
Deleted sampleimage.part.03.
...
Deleted sampleimage.part.23.
Deleted sampleimage.part.24.
Delete Bundle complete.
```

Note

If you are unable to run the AMI tools, you can use any Amazon S3 utility to delete AMIs.

You're Finished!

Congratulations! You successfully launched, accessed, and terminated an instance. For information on major Amazon EC2 features that were not covered in this guide and how to continue, see Where Do I Go from Here? (p. 45).

Please Give Us Your Feedback

Your input is important to us to help make our documentation helpful and easy to use. Please take a minute to give us your feedback on how well we were able to help you get started with Amazon EC2. Just click this Feedback Link link. Thank you.

Where Do I Go from Here?

This guide describes how to use basic features of Amazon EC2. The following describes common customer requirements and a description of where to go for more information.

The AMIs that you provide don't meet my needs

One of the first tasks that you will want to do after completing this tutorial is to create one or more custom AMIs. You can make changes to Amazon or public AMIs and modify them as needed. For more information, refer to the *AMIs* and *instances* sections of the Amazon Elastic Compute Cloud User Guide or Amazon Elastic Compute Cloud Developer Guide.

The AMI I launched during this tutorial is too small

To meet the needs of different organizations and applications, Amazon EC2 instances are available in different sizes and CPU/Memory configurations. For more information, refer to the *Instance Types* section of the Amazon Elastic Compute Cloud User Guide or Amazon Elastic Compute Cloud Developer Guide.

I need static IP Addresses

Amazon EC2 provides elastic IP addresses that can be dynamically remapped to different Amazon EC2 instances. For more information, refer to the *Instance Addressing* sections of the Amazon Elastic Compute Cloud User Guide or Amazon Elastic Compute Cloud Developer Guide.

I am concerned about keeping others from accessing my instances, both inside and outside the Amazon network

In addition to the default security group, you can create other security groups to meet your security requirements. For more information, refer to the *Network Security* sections of the Amazon Elastic Compute Cloud User Guide or Amazon Elastic Compute Cloud Developer Guide.

I want to know more about support for Windows

For more information on running Windows instances, refer to the *Windows* sections of the Amazon Elastic Compute Cloud User Guide or Amazon Elastic Compute Cloud Developer Guide.

I want to build a geographically dispersed, fault tolerant architecture on Amazon EC2

Amazon EC2 enables you to 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, refer to the *Regions and Availability Zones* sections of the Amazon Elastic Compute Cloud User Guide or Amazon Elastic Compute Cloud Developer Guide.

I will be running a set of full time or nearly full time instances and want to bring down my costs

Amazon EC2 supports an additional pricing option, which 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, refer to the *Reserved Instances* sections of the Amazon Elastic Compute Cloud User Guide or Amazon Elastic Compute Cloud Developer Guide.

I need more space than is provided on the instance store, I need a permanent storage solution, or my AMI is too big to be bundled

Amazon Elastic Block Store (Amazon EBS) 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, refer to the *Amazon EBS* sections of the Amazon Elastic Compute Cloud User Guide or Amazon Elastic Compute Cloud Developer Guide.

I want a solution for monitoring my instances

Amazon CloudWatch is a monitoring service for Amazon EC2 that is designed to gather, aggregate, store, and retrieve metrics. For more information, see Amazon CloudWatch Developer Guide.

I want a solution for load balancing requests to my instances

Elastic Load Balancing offers the ability to evenly spread requests across your running Amazon EC2 instances. For more information, see Elastic Load Balancing Developer Guide.

I want to automatically scale up and down the number of instaces that I 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, see Amazon Auto Scaling Developer Guide.

Appendix: PuTTY

Topics

Introduction

Note

This section is for Windows users using PuTTY. If you are using another operating system or SSH client, you can skip this section.

PuTTY is a free SSH client for Windows. Other tools that form part of the PuTTY suite are PuTTYgen, a key generation program, and pscp, a secure copy command line tool. This guide outlines the additional steps required to use PuTTY with Amazon EC2.

Note

The different PuTTY tools are separate applications and may require multiple downloads.

Private Key Format

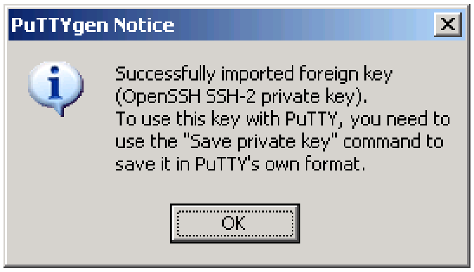
PuTTY does not natively support the private key format generated by Amazon EC2. Fortunately, PuTTY has a tool called PuTTYgen, which can convert keys to its internal format.

Note

You should have generated a private key as described in Running a Linux/UNIX Instance (p. 22) and saved the key to a file named something like `id_rsa-gsg-keypair`.

To configure PuTTY

1. Launch PuTTYgen and load `id_rsa-gsg-keypair`. PuTTYgen should pop up the following message.



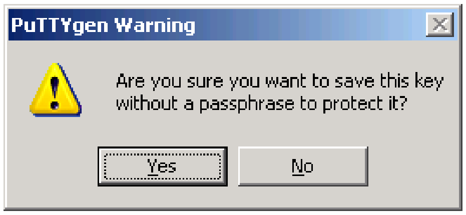
PuTTYgen Notice

Successfully imported foreign key (OpenSSH SSH-2 private key).
To use this key with PuTTY, you need to use the "Save private key" command to save it in PuTTY's own format.

Note

The private key file must end with a newline character or PuTTYgen cannot load it correctly.

2. PuTTYgen displays a lot of information regarding the key that has been loaded, such as the public key, the key passphrase, the type and the number of bits in the generated key. The keys generated by Amazon EC2 are 1024 bit SSH-2 RSA keys. They are also passphraseless. 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 is that it makes automation harder as human intervention is needed to log on to an instance, or copy files to an instance.



PuTTYgen Warning

Are you sure you want to save this key without a passphrase to protect it?

3. Save the key in PuTTY's format. You can either select **Save** from the **File** menu or click **Save private key**. Save the key as id_rsa-gsg-keypair.ppk. When PuTTYgen prompts you to save the key without a passphrase, click **Yes**.

   The file can be used with PuTTY to connect to your Amazon EC2 host as described in the next section.

SSH with PuTTY

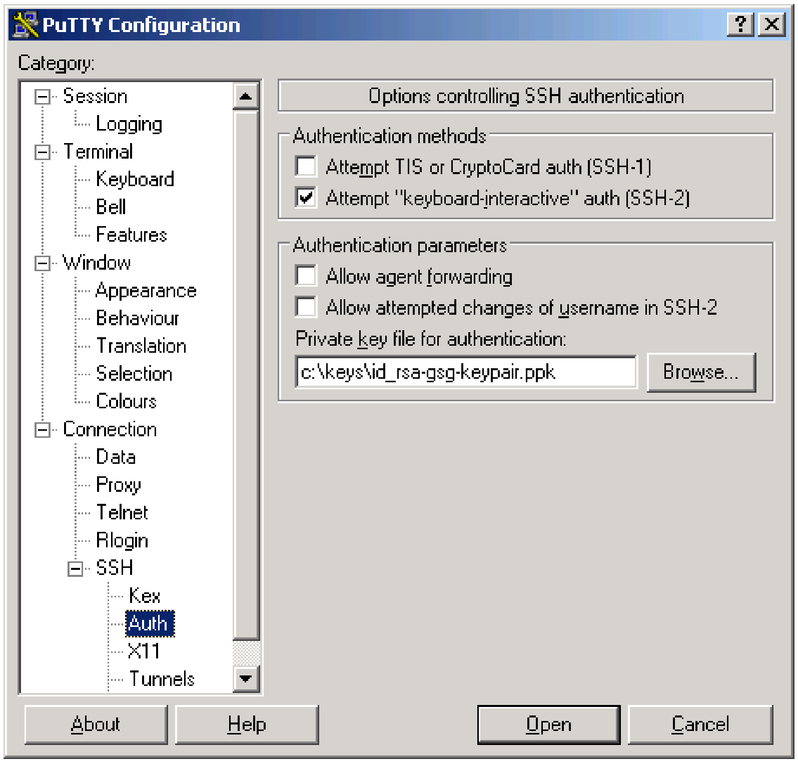
This section assumes that you have converted an Amazon EC2 generated private key file to a PuTTY private key file and have successfully launched an instance.

To use SSH with PuTTY

1. Start PuTTY. A graphical configuration utility opens.

   Click **Connection**, point to **SSH**, and select **Authentication**. The **PuTTY Configuration** dialog box appears.

   Click **Browse**, and select the PuTTY private key file you generated earlier. If you are following this guide, the file is named id_rsa-gsg-keypair.ppk.



Options controlling SSH authentication

Authentication methods

☐ Attempt TIS or CryptoCard auth (SSH-1)

☑ Attempt "keyboard-interactive" auth (SSH-2)

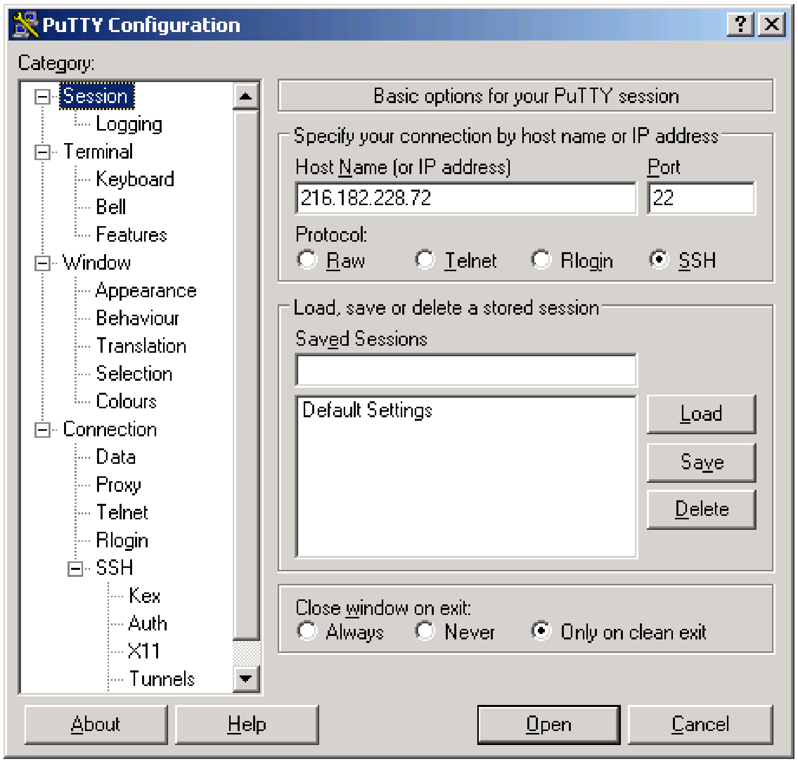
Authentication parameters

☐ Allow agent forwarding

☐ Allow attempted changes of username in SSH-2

Private key file for authentication: c:\keys\id_rsa-gsg-keypair.ppk

2. Under **Session**, enter `root@hostname` or `root@ip_address`. Click Open to connect to your Amazon EC2 instance.



Basic options for your PuTTY session

Specify your connection by host name or IP address

Host Name (or IP address): 216.182.228.72

Port: 22

Protocol: ○ Raw ○ Telnet ○ Rlogin ● SSH

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

SCP with PuTTY

The use of pscp is nearly identical to scp.

To use pscp

1. Convert your private key to PuTTY's format. The command to copy the private key and X.509 certificate, as shown in Preparing for Bundling (p. 30), should look like the following example.

   ```
   $ scp -i id_rsa-gsg-keypair pk-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem cert-
   HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem
    root@ec2-72-44-33-55.compute-1.amazonaws.com:
   ```

2. To run the same command with pscp, using the private key in PuTTY's format as generated in the preceding section, the command should look like the following example.

   ```
   C:\> pscp -i id_rsa-gsg-keypair.ppk pk-
   HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem cert-
   HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem
    root@ec2-72-44-33-55.compute-1.amazonaws.com:
   ```

Document Conventions

This section lists the common typographical and symbol use conventions for AWS technical publications.

Typographical Conventions

This section describes common typographical use conventions.

Convention	Description/Example
Call-outs	A call-out is a number in the body text to give you a visual reference. The reference point is for further discussion elsewhere. You can use this resource regularly.❶
Code in text	Inline code samples (including XML) and commands are identified with a special font. You can use the command `java -version`.

Code blocks	Blocks of sample code are set apart from the body and marked accordingly.
	`# ls -l /var/www/html/index.html`
	`-rw-rw-r--  1 root root 1872 Jun 21 09:33 /var/www/html/index.html`
	`# date`
	`Wed Jun 21 09:33:42 EDT 2006`

Emphasis	Unusual or important words and phrases are marked with a special font. You *must* sign up for an account before you can use the service.
Internal cross references	References to a section in the same document are marked. See Document Conventions (p. 51).
Logical values, constants, and regular expressions, abstracta	A special font is used for expressions that are important to identify, but are not code. If the value is `null`, the returned response will be `false`.

Convention	Description/Example
Product and feature names	Named AWS products and features are identified on first use. Create an *Amazon Machine Image* (AMI).
Operations	In-text references to operations. Use the `GetHITResponse` operation.
Parameters	In-text references to parameters. The operation accepts the parameter *`AccountID`*.
Response elements	In-text references to responses. A container for one `CollectionParent` and one or more `CollectionItems`.
Technical publication references	References to other AWS publications. If the reference is hyperlinked, it is also underscored. For detailed conceptual information, see the *Amazon Mechanical Turk Developer Guide*.
User entered values	A special font marks text that the user types. At the password prompt, type **`MyPassword`**.
User interface controls and labels	Denotes named items on the UI for easy identification. On the **File** menu, click **Properties**.
Variables	When you see this style, you must change the value of the content when you copy the text of a sample to a command line. % ec2-register <*`your-s3-bucket`*>/image.manifest See also Symbol Conventions (p. 53).

Symbol Conventions

This section describes the common use of symbols.

Convention	Symbol	Description/Example
Mutually exclusive parameters	(Parentheses | and | vertical | bars)	Within a code description, bar separators denote options from which one must be chosen. `% data = hdfread (start | stride | edge)`

Optional parameters
XML variable text
	[square brackets]	Within a code description, square brackets denote completely optional commands or parameters.
		`% sed [-n, -quiet]`
		Use square brackets in XML examples to differentiate them from tags.
		`<CustomerId>[ID]</CustomerId>`

Variables	<arrow brackets>	Within a code sample, arrow brackets denote a variable that must be replaced with a valid value. `% ec2-register <your-s3-bucket>/image.manifest`