EC2 Administrator’s Guide

# Infrastructure Services

## Elastic Compute Cloud (EC2)

It is easy to get started with the Elastic Compute Cloud (EC2). Go to aws.amazon.com/ec2/ and select “Sign Up for Amazon EC2” button to create an account. This same web page also describes EC2 and EC2 pricing in some detail.

## Simple DB

An alternative to a complex and expensive relational database, Simple DB is a simple interface to create and store, and index multiple tables to provide a way to query these tables. You can select, sort, query in real-time. Currently the limit is 100 tables (<10 GB each) with up to 256 columns. You can request larger limits.

## Simple Storage Service (S3)

The Simple Storage Service is a backing store for your cloud data. You can save execution images called Amazon Machine Images (AMI) in S3. You can also store snapshots of an AMI, databases, documentation, etc. Transfer rates between S3 and an AMI is fast (~10 MB/sec). Transfer rates between a local computer and an EC2 instance is limited to the maximum upload/download speeds of your Internet Service Provider (ISP) (~1 MB/sec).

## Simple Queue Service (SQS)

SQS is a store and forward mechanism for the transfer of messages between computers on the internet, even if they are temporarily off-line. Any computer on the internet can add or read messages. A message can contain up to 8 KB of text in any format. A message can linger in the queue for up to 4 days. SQS messages typically conform to a SOAP interface standard.

## CloudFront

CloudFront is a web interface for delivering files stored in S3 to web users. It allows public access to files stored in S3 via a web browser.

# Developer Resources

Use your web browser to navigate to aws.amazon.com/ec2/ and scroll down to Developer Resources to several links. The subsections below are a synopsis of selected links in this section. The AWS Management Console is described in the next section.

## Documentation

Select the link to the latest technical documentation set at the top of the list of links. It includes a Getting Started Guide, Developer Guide, and a Quick Reference Card. These guides are essential reading.

## Sample Code

This section under Developer Resources contains links to applications written in ColdFusion, VB, C#, Perl, Ruby, Java and others. Of particular importance is Scalr, which is a redundant, self-curing, self-scaling EC2 hosting environment. Each subsection has a download button in the upper right corner to fetch the code.

## Developer Tools

Developer Tools is found in the Developer Resources section of the aws.amazon.com/ec2/ page. A smaller set of code samples that is similar to the previous section. The Javascript Scratchpad may be of some interest. It is an alternative to the AWS Management Console that shows the properties of the instances of your AMIs running on the cloud. As with the above section, click on the download button in the upper right corner of the page to fetch the code described in each subsection.

## Articles

The Articles link is also found in the Developer Resources section of the aws.amazon.com/ec2/ page. This section includes a set of “how-to” articles aimed at specific circumstances such as running Microsoft’s SQL Server on the cloud using EBS (Elastic Block Store).

## Amazon Machine Images

The Amazon Machine Images link is found in the Developer Resources section of the aws.amazon.com/ec2/ page. Amazon Machine Images (AMIs) can be made public for others to use. Many publically available AMIs for a variety of applications can be found here. For basic machine images provided by Amazon for building an infrastructure to run a new application, see the Getting Started Guide.

# Getting Started

Use your web browser to navigate to aws.amazon.com/ec2/ and scroll way down to Developer Resources and select Documentation. Click on the first link to the latest release of Amazon EC2 and then click on View the Getting Started Guide in HTML. Follow the directions in this section to set up an account and to configure an AMI. A brief overview of the sections is included here. The comments below apply to installing the AMI tools on a Windows client and then setting up Fedora Core 8 Linux EC2 server instances on the cloud.

## Prerequisites

### AWS Account

An AWS account must be established before using the Amazon Elastic Compute Cloud (EC2). At the time that an account is set up, you can also select an account ID, access key ID, secret key, and X.509 certificate. Be sure to save your certificate file and your private key file in C:\ec2. You will need these when you set the environment variables below. You can recover your X.509 certificate, but it is not possible to recover your private key file. You will have to generate a new private key if you lose the pk-\*.pem file. For FIMCO, the User Id and password, keys, and certificate are found in S:\IT\cloud\AccountId.txt.

### Java Runtime Environment (JRE)

The Java Runtime Environment must be installed on the Windows client. The environment variable, JAVA\_HOME must be set to the path to the JRE executable as noted below.

### Environment Variables

Assuming that the account files are at C:\ec2 and that the JRE is installed as described the Getting Started Guide, a sample of the requisite environment variables are as follows:

EC2\_CERT=C:\ec2\cert-73CSPXWZ3Y4G2Y7QBAQSNPW2LJFVTOGY.pem

EC2\_HOME=C:\ec2\ec2-api-tools

EC2\_PRIVATE\_KEY=C:\ec2\pk-73CSPXWZ3Y4G2Y7QBAQSNPW2LJFVTOGY.pem

JAVA\_HOME=C:\Program Files\Java\jre1.5.0\_17

## Setting up the Tools

After downloading the command-line tools as noted in the guide, set the EC2\_HOME environment variable to point to the folder containing the tools and add the path to the %EC2\_HOME%\bin folder to the PATH environment variable. Do not set the EC2\_URL environment variable as described. The URLs shown are no longer valid. If you have everything setup properly, you can open a command-line (DOS) window and execute any of the ec2 commands provided. As an alternative, you can also perform some basic tasks using the AWS Management Console as described below.

## Running a Linux Instance

Select “Running an Instance” in the Getting Started Guide -> Running an Instance -> Linux and UNIX. Verify that all of the above prerequisites are installed on your Windows client. Open a command-line window and enter the following command to see a few simple public images:

ec2-describe-images –o amazon

or enter the following command to see any private images that are owned by your current account:

ec2dim –o self

The list of Amazon images includes the following entry where the AMI ID is ami-2547a34c:

IMAGE ami-2547a34c ec2-public-images/fedora-8-x86\_64-base-v1.08.manifest.xml amazon x86\_64 …

A gsg-keypair has already been created. It is at C:\ec2\id\_rsa-gsg-keypair as noted below.

A set of Ubuntu images can are also available at alestic.com. There is a selection of images that include a desktop and a skinnier set of “server” images that are smaller and faster to start up.

### Key Pair

An SSH Key Pair may be generated from a local console window by entering the command:

ec2-add-keypair gsg-keypair

Create a file named id\_rsa-gsg-keypair in C:\ec2 and then paste the entire RSA Private Key into this text file. Include the header and the footer in the file. On a Windows client, convert this private key to PuTTY’s format. See the Appendix: PuTTY in the EC2 Getting Started Guide for details.

### Instance Types

A complete list of instance types is found in the Developers Guide -> Using Amazon EC2 -> Launching and Using Instances -> Instance Types. Use only m1.large, m1.xlarge, or c1.xlarge with a 64-bit AMI and use only m1.small and c1.medium with 32-bit AMI.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **CPU** | **Memory** | **Storage** |
| m1.large | 2 cores (2 EC2 each) | 7.5 GB | 2 x 420 GB |
| m1.xlarge | 4 cores (2 EC2 each) | 15 GB | 4 x 420 GB |
| c1.xlarge | 8 cores (2.5 EC2 each) | 7 GB | 4 x 420 GB |
| m1.small | 1 core (1 EC2) | 1.7 GB | 1 x 160 GB |
| c1.medium | 2 core (2.5 EC2 each) | 1.7 GB | 1 x 350 GB |

### Launch Instance

Launch an instance of this public AMI using where the instance type is one of the above names:

ec2-run-instances ami-2547a34c –k gsg-keypair –t m1.large

Note the instance id (e.g. i-1303977a) which will be used to identify this instance. Be sure to terminate the instance when it is no longer being used to avoid incurring unnecessary charges!

Port 22 (SSH) and port 80 (HTTP) should already be authorized. If not, authorize port 22 (SSH) and port 80 (HTTP) access to this instance using:

ec2-authorize default –p 22

ec2-authorize default –p 80

### Installing a Windows SSH Client

In order to establish a secure remote connection to this instance of the AMI, an SSH client must be installed on your copy of Windows. Follow the directions provided to install PuTTY in the Getting Started Guide -> Prerequisites -> Accessing Windows Instances through SSH clients. Navigate to http://www.chiark.greenend.org.uk/~7tatham/putty/ in your web browser and then click on Download link. You will need to copy putty.exe, puttygen.exe, and pscp.exe for Windows into %EC2\_HOME%\bin. Follow the directions in the Getting Started Guide -> Appendix: PuTTY to convert the id\_rsa-gsg-keypair into PuTTY format. If id\_rsa-gsg-keypair was generated on a Linux machine, first convert the line terminations in this file to CRLF. You can edit the file with aised.exe to convert the line terminations.

### Access an Instance

If the image includes a web server, you may view the web pages being served using your favorite web browser. The fourth column of the output from the following command shows the URL assigned to this instance:

ec2-describe-instances

or use the short name ec2din. You can view the console output generated on startup, using:

ec2-get-console-output i-1303977a

Finally, you can use the procedure described in the Getting Started Guide Appendix: PuTTY to instantiate a remote bash shell if on your machine. The ssh command that is shown in the Running an Instance will not work on a Windows client. Enter the following command in a command window on your local machine:

putty

In the PuTTY Configuration dialog, expand SSH -> Auth and enter the path to the private key file for authentication:

C:\ec2\id\_rsa-gsg-keypair.ppk

Highlight Session and enter the host name:

root@ec2-xxx.compute-1.amazonaws.com

Replace xxx with the URL shown by ec2-describe-instances for this instance. Click on Open to bring up a remote bash shell that is connected to your new instance.

## Bundling an AMI

### Upload Files to Instance

It is possible to modify a basic instance such as the one created above and then save it in S3 as a private AMI that is configured to run an application of your choosing. Refer to the Getting Started Guide -> Running an Instance -> Linux & UNIX -> Bundling an AMI for more detail. You can upload files from your Windows client to an instance on EC2 using pscp that you downloaded from the PuTTY site. To upload one or more files, in a local command window use

cd \ec2

pscp –i id\_rsa-gsg-keypair.ppk files… root@ec2-xxx.compute-1.amazonaws.com:/path

Replace xxx with the URL for your particular instance and replace path with the path to the destination in the instance’s file system. The xxx portion of the URL is, in fact, the IP address assigned to the instance. For example, to upload the private key and X.509 certificate to this AMI instance, replace the command shown in the Getting Started Guide with the following command:

pscp -i id\_rsa-gsg-keypair.ppk pk-\*.pem cert-\*.pem root@ec2-xxx.compute-1.amazonaws.com:/mnt

Replace xxx with the URL for your particular instance. Run the command ec2din to get the current URL for your instance. If you are asked for a password, the key was not available or one or more minus signs were not interpreted as an ASCII minus when you pasted from the above line. Retype all the minus signs in the command in the console and re-run the command.

### Bundling

Refer to the Getting Started Guide -> Running an Instance -> Linux and UNIX -> Bundling an AMI for more detail. To prepare an AMI of this instance that can be saved in S3, start up a remote bash shell as described above under Access an Instance (commands to be entered in a remote bash shell are denoted by a # prefix).

# ec2-bundle-vol -d /mnt -k /mnt/pk-\*.pem -c /mnt/cert-\*.pem -u 853885810817 -r x86\_64 -p AisDevUb8.04

where the –u argument is your AWS account ID, the –r argument is the instance type and the –p argument is the name of the AMI that you are saving.

### Upload Image to S3

Refer to the Getting Started Guide -> Running an Instance -> Linux and UNIX -> Bundling an AMI for more information on uploading an image to S3. To upload the bundled image, enter the following command in your remote bash shell:

# ec2-upload-bundle -b AisImage -m /mnt/AisDevUb8.04.manifest.xml -a 0K41DJKQ7SB6EDHEHHG2 -s wxxX/VEhOw3K1JDmbWsS7PlPiimw2AK7mC4yErUS

Where the –b argument is the S3 bucket name, the –a argument is your Access Key ID and the –s argument is your Secret Access Key.

### Register the AMI

Finally, register this AMI to get an AMI identifier. From a command window on your local computer, use

ec2-register AisImage/AisDevUb8.04.manifest.xml

to get the new AMI identifier ami-3c739455. If this AMI has already been registered, you must first deregister the existing image (see next section) and the register it again.

### Deregister the AMI

Be sure to remove any AMI that is not being used to save on S3 charges. To remove an AMI from S3, deregister it using:

ec2-deregister ami-3c739455

and then use the AWS Management Console (see below) to delete the files associated with that AMI from S3.

### Terminate the Instance

From a command window on your local computer, be sure to terminate the executing instance:

ec2-terminate-instances i-1303977a

where the instance ID is obtained from ec2din command.

# AWS Management Console

## Overview

The Amazon Web Services (AWecS) Management Console provides a global view of the AMI instances currently in-execution. It also allows you to manage these instances as follows:

* Start or stop an EC2 instance.
* View, create, or delete AMIs in S3
* View, create, or delete Elastic Block Store (EBS) volumes and snapshots.
* Attach or detach volumes to EC2 instances
* Configure firewall settings. Manage public and private access keys.

## Future Plans

Several additional features are planned for the AWS Management Console:

* **Tags.**  Label and group EC2 resources (AMI, volumes, databases) with custom metadata (tags).
* **Load Balancing.** Monitor and configure EC2 instance metrics. Impose auto-scaling rules to balance the load.
* **S3 Support.** Create or delete buckets. Upload or download files to S3.
* **Simple DB Support.** Create or edit queries. View datasets.
* **SQS Support.** Manage queues. Add or retrieve messages from existing queues.
* **CloudFront Support.** Setup and test web delivery of content from S3.

## Web Browsers

AWS Console works with the following web browsers:

* Mozilla Firefox 2.0 or greater
* Internet Explorer 7.0 or greater
* Safari 3.0 or greater

If you are still using IE 6.0, you will have to upgrade.

## Management Console Tour

The AWS Console has a few prerequisites. You must have an Amazon AWS account set up with the appropriate security codes established. See the Getting Started Guide described in the Documentation section above for more information on setting up an account and the appropriate security keys.

One way to start the AWS console is to navigate to aws.amazon.com/ec2/, scroll way down to Resources and select AWS Management Console. You can go to this page directly by navigating to <https://console.aws.amazon.com/> . If you are logged in to our account, you can select the Amazon EC2 tab to launch the Amazon EC2 Console. The following sections note some typical operations.

### Launch Instances

Select the Launch Instances button on the EC2 Console. Select My AMIs to view the images of interest. Select the Select button on the right to launch an image that is stored in S3. This will create an instance of that AMI running on the cloud. Perform the following step in the Launch Instances dialog.

1. Number of instances: 1
2. Instance type: m2.large
3. Key Pair Name: gsg-keypair
4. Security Groups: default

Select the Launch button at the bottom of the dialog. Select the Close button at the bottom of the dialog. On return to the EC2 Dashboard, select the Refresh button in the upper right corner of the dashboard.

### View Instances

Select Instances link in the left side panel or select the Running Instance in the My Resources group. You can return to the Dashboard at any point by selecting the EC2 Dashboard link at the top of the left side panel. To view more detail on the running instance, check the box at the left of the running instance. The instance properties are shown below the list of instances. The buttons in the toolbar at the top allow you to Launch Instances, Reboot, or Terminate, or Connect to an instance. The Output toolbar icon shows the standard output generated by the selected instance. It is quite informative.

**Terminate.**  Be sure to terminate the instances that you launch when you are finished to avoid running up an unnecessary charges on your AWS account.

### AMIs

Select the AMIs link in the left side panel of the Dashboard. By default All Images are selected. In the Viewing drop-down menu at the top of the right panel, select “Owned By Me”. You can select an AMI from the list and then select the Launch button in the toolbar at the top. An instance of the image is uploaded to a server on the cloud and started up. At this point you can view the output, connect to this instance or terminate this instance as noted in the previous section on View Instances.

# AIS Images

There are two basic images that are described here. The first is a slimmed down image that just runs AIS as a service. It contains aissvc.exe and the shared libraries that are required to run this service. Users may access an instance of this AMI using RIDE. Users may supplement this basic image with aisLisp code that targets their specific application. The second image is a complete AIS development environment.

**AI Server Image.**  This is a set of images designed for various applications.  It includes the libraries and executables required to run aissvc. It does not include X11 or any desktop support.  Each image may have a collection of contexts that are tailored to a specific range of applications.  Variations of this image may include some specialized aisLisp code that has been contributed by the open source community (especially from Universities).

**AIS Developer Image.** This image includes all the stuff that is described in Frank's document on the SourceForge wiki, AisEclipseQT for Linux.  This image saves developers hours of effort to setup and configure a development environment. It saves us endless questions about why a hand-crafted development environment does not work.  Instead, users can just install an NX client on their local laptop to access an instance of our publically available AMI and do their development on the cloud.  This AMI includes the following:

* Ubuntu 8.0.4 x\_64 including desktop
* NX Server
* QT 4.3.4 for Linux
* Eclipse IDE
* Embedded MySQL 5.1.30
* AisDev 4.003
* VirtualBox 2.1.4

## AI Server Image

This section is very specialized information for constructing an AMI of this type. It is only of interest if you are involved in building this AMI. This section is not of interest to users. They should refer to the wiki for help on using this AMI.

## AIS Developer Image

This section is very specialized information for constructing an AMI of this type. It is only of interest if you are involved in building this AMI. This section is not of interest to users. They should refer to the wiki for help on using this AMI.

# NFS Setup

NFS (Network File System) was created by Sun Microsystems in 1985 as a competitor to DECnet developed in the 1970s by Digital Equipment Corporation. Today, NFS is widely used by Linux-based machines to allow a NFS client to access a portion of a file system on a NFS server. Any one computer can be an NFS client, a NFS server, or both. The remote NFS appears to the client just as though it is a local file system (except that updates are slower). The following provides guidance on setting up an NFS server and a NFS client on an Ubuntu-based machine. It is distinguished from a multitude of existing installation documentation in that it describes what to do when the instructions don’t work.

## NFS Server Setup

If the NFS Server is not installed, it can be installed using the aptitude package manager. The appropriate packages are fetched from us.archive.ubuntu.com and security.ubuntu.com (see /etc/apt/sources.list for more details). All of the following commands are run in a remote bash shell (see Installing a Windows SSH Client above). Commands are prefixed with an octothorpe (#) to indicate that they are run as root from a remote bash shell.

**NFS Server Status.** To see if the NFS server is already installed, enter the command:

# apt-list nfs-kernel-server

If apt-list is not available, it may be installed using:

# apt-get install apt-file

Notice the absence of any dashes in front of the install option. If apt-list still does not work, you may need to update, using:

# apt-file update

**NFS Server Install.**  To install NFS Server on an Ubuntu instance, enter the command:

# apt-get install nfs-kernel-server

Notice that no dashes are allowed in front of the install option. If this command fails, perform an update using:

# apt-get update

**NFS Server Configure.** One line must be added to three files in order to configure NFS. Each file is listed below with the line to be added.

/etc/default/nfs-common STATDOPTS=”--port 32765 –outgoing-port 32766”

/etc/default/nfs-kernel-server RPCMOUNTDOPTS=”-p 32767”

/etc/exports /greenteam 10.0.0.0/255.0.0.0(rw,async,wdelay,no\_subtree\_check)

The line added to exports allows all NFS clients in EC2 to access the /greenteam directory on this server. These NFS clients have read and write access. Async allows a delayed write (reply before write completes). Wdelay causes the server to delay writes to the disk in anticipation of multiple write requests in a single operation. No\_subtree\_check skips the check to make sure that the file exists in the /mount subtree. If a file is renamed while it is opened, the subtree check would cause the operation to fail. If you wish to allow the superuser on the client to be able to modify files that are owned by root in /greenteam, add the “no\_root\_squash” option to the above options in the exports file.

Issue the following command to cause the exports file to be reread:

# exportfs –r

After configuring NFS call the init script to reexport the exported directories and restart the nfsd daemon:

# /etc/init.d/nfs-kernel-server restart

**NFS Server Status.** To check the NFS server status, use the first command to make sure that the NFS service is running:

# /etc/init.d/nfs-kernel-server status

nfsd is running

Also, check that mountd is active:

# ps –eF | grep mountd

29609 00:00:00 rpc.mountd

Use rpcinfo to make sure that NFS is registered with portmap:

# rpcinfo –p localhost | grep nfs

## NFS Client Setup

If the NFS Client is not installed, it can be installed using the aptitude package manager. The appropriate packages are fetched from us.archive.ubuntu.com and security.ubuntu.com (see /etc/apt/sources.list for more details). All of the following commands are run in a remote bash shell (see Installing a Windows SSH Client above). Commands are prefixed with an octothorpe (#) to indicate that they are run as root from a remote bash shell.

**NFS Client Status.** To see if the NFS client is already installed, enter the command:

# apt-file list nfs-common

If apt-file is not available, it may be installed using:

# apt-get install apt-file

Notice the absence of any dashes in front of the install option. If apt-file still does not work, you may need to update, using:

# apt-file update

**NFS Client Install.**  To install NFS client on an Ubuntu instance, enter the command:

# apt-get install nfs-common

Notice that no dashes are allowed in front of the install option. If this command fails, perform an update using:

# apt-get update

**NFS Client Configure.** The NFS Client is automatically configured when it is installed. To check that the necessary rpc process are running, enter the command on the NFS Client console window:

# ps –eF | grep rpc

The rpc.statd, rpc.idmapd, and rmpc.mountd processes should be running.

**NFS Client Mount.** In order to access the greenteam folder on the NFS Server, use the following mount command:

# mount ec2-XXX.compute-1.amazonaws.com:/greenteam /mynfs

where mynfs is the mount point on the NFS client and XXX is the URL of the NFS server. The mount point should be an empty, local directory (any files in this folder become invisible when another file system is mounted on it). A remote file system can be automatically mounted during startup by adding an entry in the /etc/fstab file. For example, add the following line to /etc/fstab on the client:

ec2-XXX.compute-1.amazonaws.com:/greenteam /mynfs nfs rsize=8192,wsize=8192 0 0

This option is a bit of a problem in that the NFS client must know the URL or the IP address of the NFS Server. If the server has been assigned a static IP address, then this entry can be specified across multiple instances of the server. Increasing the rsize and wsize options will improve performance, especially if the two machines have a high-speed connection. The final two zeros prevent any backup or file check by the client of the remote file system (backups and file system checks are done by the server).

# Installing Rpm Packages

An RPM (RedHat Package Manager) file is designed for Redhat-based systems such as Redhat, openSUSE, Fedora, Mandriva, etc. In Debian-based systems, such as Ubuntu, depend upon dpkg (Debian Package). Several tools, such as apt (advanced package tool) and aptitude are tools built on top of dpkg.

**Alien.** It is not possible to use rpm directly to install an rpm package. However, it is possible to convert an rpm file to a deb file using the application named alien. To install alien, enter the following command in the remote bash shell:

# apt-get install alien

And use the following command to convert the following rpm file to a deb file:

# alien -d ec2-ami-tools.noarch.rpm

which generates ec2-ami-tools\_1.3-31781\_all.deb.

**Install.** To install this Debian package, use:

# dpkg -i ec2-ami-tools\_1.3-31781\_all.deb

# Installing EC2 API Tools

All of the following commands are run on a remote bash shell that is connected to an instance running on the cloud. The commands are prefixed by a $ to indicate that you are running as the user named sysadmin (the password for sysadmin is sysadmin). The root password is $cuban88. If not, you can log in as sysadmin as follows:

# su sysadmin

$ cd ~

$ . .profile

where the dot command, ., is used to run the startup shell.

## Java Runtime Executable

The API Tools require the Java Runtime from Sun Microsystems. Verify that the following folder exists:

$ ls /usr/lib/jvm/java-1.5.0-sun

If it does not exist, you can install the jre package as follows:

$ sudo apt-get install sun-java5-jre

## Unzip

You can just enter the command “which unzip” into a remote bash shell to determine if unzip is already installed. If unzip is not installed, use the following command to install unzip:

$ sudo apt-get install unzip

## Download EC2 API Tools

Be aware that the API tools are separate from the AMI tools (which are normally already installed on an AMI). The API tools are those that are also installed on your local computer in order to manage your EC2 environment. They can also be installed on an instance in order to allow an instance on the cloud to launch additional worker instances.

### Download

Download the EC2 command-line tools from the Amazon Resource Center to the sysadmin home directory using:

$ cd ~

$ wget http://s3.amazonaws.com/ec2-downloads/ec2-api-tools-1.3-34128.zip

$ unzip ec2-api-tools-1.3-34128.zip

### Configuration

**Keys.** The following files should be available in /home/sysadmin/ec2/:

AccountId.txt

id\_rsa-gsg-keypair

cert-73CSPXWZ3Y4G2Y7QBAQSNPW2LJFVTOGY.pem

pk-73CSPXWZ3Y4G2Y7QBAQSNPW2LJFVTOGY.pem

If not, use pscp to copy these files from your local computer up to your instance.

**Environment.** Make sure that the following environment variables are set in /home/sysadmin/.profile:

export EC2\_HOME=$HOME/ec2-api-tools-1.3-34128

export PATH=$PATH:$EC2\_HOME/bin

export EC2\_PRIVATE\_KEY=$HOME/ec2/pk-73CSPXWZ3Y4G2Y7QBAQSNPW2LJFVTOGY.pem

export EC2\_CERT=$HOME/ec2/cert-73CSPXWZ3Y4G2Y7QBAQSNPW2LJFVTOGY.pem

export JAVA\_HOME=/usr/lib/jvm/java-1.5.0-sun/

**Test.** In order to test the installation, launch an AMI from this instance using the following commands:

$ ec2dim –o self

$ ec2run ami-XXXXXXXX -k gsg-keypair -t m1.large

$ ec2din

where the first command will show you the AMI id number that is required in the second command. In case these commands fail, you can get more information from help.ubuntu.com/community/EC2APITools .

## Installing MySql on Ubuntu

**Install.** Installing MySql on an Ubuntu-based AMI is simple. Just enter the following command in a remote bash shell on a development instance:

$ sudo apt-get install mysql-server

**MySql Group.** A MySql user and group are added automatically. Just to check and add, if necessary, enter the following commands as root on a remote bash shell:

# groupadd mysql

# useradd –gid mysql mysql

You will typically get a message that the group and user already exist.

## Configuring MySql on Ubuntu