CS162 Spring 2012 - Projects 3 and 4
EC2 Access Guide

(Specification Version 1.0)

EC2

Amazon EC2 is a cloud service that lets users start and stop virtual servers on demand, termed “instances”. These instances are based off a virtual machine image that can be specified, termed an Amazon Machine Image (AMI). You are given full and complete access to your instance (meaning root access), but since it is a virtualized platform, you might still be sharing physical resources with other EC2 users. We are using “small” EC2 instances, which means that the level of hardware multiplexing is pretty high, but enough for the minimal requirements of our projects.

CS162 has been granted a single “master” EC2 account, from which we are spinning off “subaccounts” for each student. To make this work, we’ve written a number of scripts that reside on stella.cs.berkeley.edu, which allow you to start, stop, and otherwise manage your own instances through this master EC2 account. *These scripts will also provide you with SSH keys that will allow you to log into instances that you start. You must use this SSH key to login to your EC2 instances.* Password SSH access is not enabled. A more detailed description of how to use these scripts follows under “Using Our EC2 Scripts”.

All of the CS162 instances that are started will be based off of a slightly customized AMI, which has important packages like svn, javac, vim, and screen installed. Some familiarity with command line tools is necessary to interact with your machines. You need to be able to run Java programs on the command line. You are also given root access through sudo on your own instances, so you can install additional packages and otherwise configure your instance however you wish.

Email cs162 if you have any suggestions for other packages that would be nice to have included in the base AMI. Student subaccounts cannot roll their own AMIs, but we are definitely open to suggestions on how the base one can be improved.

Operational notes

Another word on EC2 terminology. Instances can be launched, started, stopped, and terminated. Here is a summary of what these operations mean:

- Launch: start a brand new instance based on an AMI
- Stop: pause a running instance, shutting it down but saving local changes
- Start: unpause a previously stopped instance, recovering the saved state
- Terminate: stop an instance and throw away local changes. Have to launch a new AMI.

This means that if you copy your code over, work on it a bit, stop it when you're done, then start it back up again later, your code will still be there on the instance. However, if you terminate the instance, this completely wipes the instance, and you cannot recover any changes you have made. Terminating an old instance and launching a fresh instance based on the base AMI is a good way of resetting everything, if you have really messed up the instance somehow.

Using our EC2 scripts

To run our scripts, you **must** be SSH-ed into stella.cs.berkeley.edu

Follow these steps to set up your account to run EC2 instances:

1. SSH into stella.cs.berkeley.edu using your class login account.
2. Run /share/b/cs162/bin/ec2_monitor --init. This command will create an SSH key pair for you and add a .pem file to your home directory. You should only need to run --init once per class login account, but if you accidentally delete your .pem file, you can run --init again to restore it.

Once you have taken these steps, you are now ready to manage EC2 instances using your account. Each member of your group can register and manage instances independently. All management will be done using the /share/b/cs162/bin/ec2_monitor script.[1]

The --help command will show you how to use the script. Options are:

```
-h, --help                  Show this help message and exit.

-i, --init                  Initialize your class EC2 account. Run once.

-l, --list                  List instances that you have access to.

-a, --launch                Launch a new instance.

-s STOP, --stop=STOP        Stop a running instance.

-S, --stop-all              Stop all running instances.

-r RESUME, --resume=RESUME  Resume a stopped instance.

-R, --resume-all            Resume all stopped instances.

-t TERMINATE, --            Terminate a running or stopped instance.
terminate=TERMINATE

-T, --terminate-all         Terminate all of your instances.
```

[1] You may want to edit your $PATH in .bash_profile to include /share/b/cs162/bin/, but only do this if you are sure of what you are doing.

The --list option will only show instances that you have launched and not yet terminated. Any stopped instances appearing in this list may be resumed using the --resume= option with the Instance ID of the instance you wish to resume. Options for stopping and terminating instances follow a similar pattern.

To manage a particular running instance, obtain the instance's hostname by using the --list option. You can then SSH into that instance by running the following command:

```
ssh -i <keyfile> ec2-user@<hostname>
```

where <keyfile> is the filename of the .pem file created when you ran --init, and <hostname> is the instance's hostname.

Also note that it can take a minute or two to do these operations. Give the instance a minute or two to boot before deciding you can't SSH in.

Here is a a short example sequence of commands to ec2_monitor, in which we list instances, resume a stopped instance, and login to it.

```
stella [501] ~ # /share/b/cs162/bin/ec2_monitor -l
Instance ID   State          Uptime   Hostname
i-9d840efe    stopped        45:26
```

In this example, instance i-9d840efe is stopped, so we first resume it.

```
stella [508] ~ # /share/b/cs162/bin/ec2_monitor -r i-9d840efe
Attempting to resume instance i-9d840efe...
Done
```

We allow a minute or two after seeing this output for Amazon Web Services to actually get the instance back up. We then verify that it is running.

```
stella [510] ~ # /share/b/cs162/bin/ec2_monitor -l
Instance ID   State          Uptime   Hostname
i-9d840efe    running        00:00    ec2-107-22-0-156.compute-
1.amazonaws.com
```

Its hostname is ec2-107-22-0-156.compute-1.amazonaws.com. We proceed to login to it.

```
stella [511] ~ # ssh -i cs162-kl-default.pem ec2-user@ec2-107-22-0-
156.compute-1.amazonaws.com
```

EC2 billing

Using EC2 costs money. We have been allocated enough EC2 credit that this should not be a

problem, but we are depending on all of you to be responsible with your usage. This basically boils down to not starting a large number of instances and then leaving them on overnight when you're not using them, but let us break it down a little further

Instances are charged at an *hourly rate* while they are started. The minimum charge per instance is also a single hour. Instances incur negligible cost when they are stopped, and zero cost after they have been terminated. This means a few simple rules can keep our costs in line and everyone happy:

- **Do not start a large number of instances.** 2-5 is okay, 20 is not. We will not be testing with 20 instances, and it is rather unnecessary.
- **Do not leave instances on when you are not using them.** Stop them if you care about state, or just terminate and blow them away.
- **Make sure your instances are stopped or terminated before logging off.** It takes some time.
- **Do not start and stop instances frequently in a short timespan.** Every time you start an instance, it charges a minimum of 1 hour of usage, so this behavior can become expensive quickly.

We are also considering providing accounting scripts to help you monitor your own usage, or configuring instances to automatically stop themselves after a few hours to prevent any "I forgot and left for the weekend" type situations. However, strict enforcement and budgeting should not be necessary as long as everyone is careful about their usage.

Hints

Using EC2 Effectively

- When working remotely, at some point you will probably want to start a session, run some commands, log out, and resume the same session again later. screen is a useful tool for this purpose, and we recommend that you learn how to use it for this project. Here is a very simple tutorial on how to use screen and which commands are most important to know: http://www.mattcutts.com/blog/a-quick-tutorial-on-screen/

Specification Changelog

Version 1.0

- Initial release