

LineUp

Team Aphrodite

Nora Allison (efa91)

Bryan Djunaedi (dju90)

Stephen LaPlante (laplansk)

Evan Leon (ejl6)

Thomas Rothschilds (tgr4)

Simone Schaffer (simone09)

Nicholas Shahan (nshahan)

Evan Whitfield (evanw2)

*Production Setup
Instructions*

Starting an Amazon EC2 instance

Options on the same page or pane are grouped together.

1. Open up the EC2 dashboard for the US-West (Oregon) availability zone.
2. Click Launch Instance
3. Choose the host operating system 'Ubuntu Server 14.04 LTS (PV)', with 64 bit selected.
4. Check that the Instance Type is micro (unless you wish to provision a larger instance)
5. Select 'No Preference' for the launch subnet.
6. Leave monitoring, user data, public IP, IAM role, shutdown behavior and tenancy in their default state.
7. Enable termination protection.
8. Select the default Kernel and RAM Disk IDs
9. Use the default storage device configuration, unless you have some specific reason to change it (e.g. would like to provision more disk space, and are allowed to).
10. Don't add any tags.
11. Either create a new security group that allows SSH connections (this is the default security group settings), or use an existing security group that you know to be configured correctly. You may want to add inbound HTTP access as well.
12. Click 'Launch'. Either generate a new keypair, or use an existing one. This keypair should only be used for our 403 project, and not for any personal EC2 instances you may have, since you may need to share it with other team members.
13. Make sure you have downloaded the keypair as a .pem file. Ideally, install this file into your SSH configuration by copying it to ~/.ssh and running chmod 600 on the file.

Viewing provisioned EC2 instances

1. Open the EC2 Management Console for the US-West availability zone.
2. Click on the Instances link in the menu on the left
3. To view details about an instance, click on it in the table. This data includes
 - a. The internal and external IP of the instance
 - b. The domain name provided by Amazon for the instance
 - c. Links to modify or terminate the instance.

Accessing an EC2 instance as root

This should only be used if individual user accounts are unavailable or broken. **Whenever possible, use sudo to impersonate root instead of logging in as a root user.**

If your keypair is installed, run the following command to ssh into an EC2 instance:

```
ssh -i ~/.ssh/[keypair name].pem ubuntu@[instance domain or IP]
```

The root user is also accessible, however its use is discouraged. Ubuntu Server comes pre-configured with an ubuntu user that is a sudoer.

Configuring user permissions

Create an individual user

```
sudo adduser [username]
```

You will be prompted for a password, name, and other miscellaneous information that can be left blank.

Add a user to sudoers

```
sudo adduser [username] sudo
```

This adds the given user to the group 'sudo', which is configured to grant sudo access to its members. **Do not manually edit /etc/sudoers unless absolutely necessary.**

Enable SSH password authentication

By default, EC2 instances are configured to reject password authentication for SSH, and require a keypair instead. Though more secure, this is inconvenient and overkill for our needs. To enable password authentication:

1. Open /etc/ssh/sshd_config and set PasswordAuthentication to 'yes'.
2. Reload sshd's configuration data by running `sudo service ssh reload`

Once SSH password authentication has been enabled, log out of the ubuntu user and into your personal user. **Use your personal user when accessing the server unless you have a specific reason for using ubuntu or root.**

Setting up Apache

Installing Apache

```
sudo apt-get install apache2
```

Starting, stopping, restarting or reloading Apache

```
sudo service apache2 status
sudo service apache2 start
sudo service apache2 stop
sudo service apache2 restart
etc...
```

Restarting Apache forces a re-compile of WSGI, and by extension Flask. It can also help resolve any glitches that crop up when editing Apache's configuration on-the-fly.

Miscellaneous Information on Apache

- Apache's configuration files live in /etc/apache2
- Apache's logs live in /var/log/apache2.

Installing Python

Python will already be installed with the Ubuntu distro described above. If you chose a different distribution, install Python 2.7.6 with your distribution package manager.

Installing Flask (and pip, if necessary)

If pip is not already installed, you will need it to install flask.

Download the pip python file [here](#) (more detailed instructions can be found [here](#) if necessary)

Run the following line:

```
python get-pip.py
```

Install flask using pip:

```
pip install Flask
```

Configure Apache/WSGI to work with Flask

- see instructions on flask.pocoo.org/docs/deploying/mod_wsgi
 - at step “Configuring Apache” do not include a user nor group the `wsgiDaemonProcess` declaration.
 - Final `[application].conf` file should be as follows and within `/etc/apache2/sites-enabled`:

```
<VirtualHost *:80>
    #ServerName lineup.com

    ServerAdmin tgr4@uw.edu
    DocumentRoot /var/www/lineup/app/static
    WSGIDaemonProcess lineup python-path=/var/www/lineup threads=50
    WSGIScriptAlias / /var/www/lineup/lineup.wsgi

    <Directory /var/www/lineup>
        WSGIProcessGroup lineup
        WSGIApplicationGroup %{GLOBAL}
        Order deny,allow
        Allow from all
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/lineuperror.log
    LogLevel warn
</VirtualHost>
```

Remove the default `.conf` symlink from `/etc/apache2/sites-enabled` (probably named `000-default.conf`)

Restart Apache Server

```
$sudo service apache2 restart
```

Installing Git

```
sudo apt-get install git
```

Checkout the LineUp repository

Navigate to the /var/www folder.

Clone the repository from the branch that you would like to install github.

```
sudo git clone https://github.com/Line-Up-Admin/CSE-403-Spring2014.git -b  
[Desired version release] lineup
```