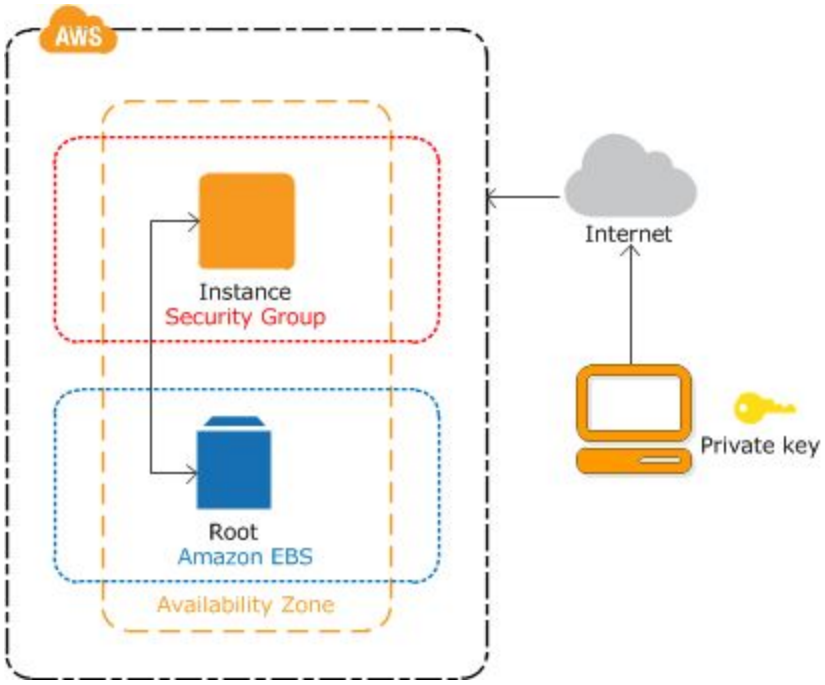
Server Set Up Documentation

By Purity Maina

1) Create an aws instance.
2) Connect to your instance
3) Generate a key
4) Add key to bitbucket/any available version control tool.
5) Clone repo
6) Set up (python, database(postgres),server(apache))
7) Create a database
8) Load fixtures
9) Migrate
10) Link wsgi to apache.

Create an aws instance.



You can launch a Linux instance using the AWS Management Console,
Step by step guide is found here :
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html#ec2-launch-instance_linux

Connect to Your Instance

Having launched an instance from an Amazon Linux AMI which has a specific user name,you can connect using SSH client

1. From the Amazon EC2 console, choose Instances in the navigation pane.
2. Select the instance, and then choose Connect.

Alternatively,conect using your key and a username from the terminal.

1. Get/Download the .pem file.
2. Save it to a specific folder.
3. Cd to that folder.
4. Connect via ssh
   Eg ssh -i name.pem user@instance.compute.amazonaws.com

Genearate Key

https://www.digitalocean.com/community/tutorials/how-to-set-up-ssh-keys--2

Create the RSA Key Pair;ssh-keygen -t rsa

Store the Keys and Passphrase:Enter file in which to save the key (/home/demo/.ssh/id_rsa):

Add key to bit bucket

- ➢ Settings ; add keys

Clone repo

- ➢ Git clone url

Set up (python, database(postgres),server(apache))

https://www.digitalocean.com/community/tutorials/how-to-serve-django-applications-with-apache-and-mod_wsgi-on-ubuntu-14-04

- ➢ Python -- sudo apt-get install python-pip apache2 libapache2-mod-wsgi
- ➢ Django -- sudo apt-get install python-django
- ➢ Postgres -- sudo apt-get install postgresql
- ➢ Apache -- sudo apt-get install apache2
- ➢ Configure a Python Virtual Environment -- sudo pip install virtualenv
- ➢ Pip -- sudo apt-get install python-pip

Check Installations

python --version

django-admin --version

- **Steps to Install database**Install Postgres -- sudo apt-get install postgresql
- Log into Postgres --- sudo -u postgres psql postgres
- Add password (default user does not have a password) -- ALTER USER Postgres WITH PASSWORD '<newpassword>';

Create a database

- Createdatabase lab;

Migrate

- Install and Freeze Other Requirements (Optional)--pip install -r requirements.txt
- Python manage.py make migrations
- Python manage.py migrate

Load fixtures

- Create JSON fixtures
- Call manage.py loaddata <fixturename>, where <fixturename> is the name of the fixture file you've created.

Link wsgi to apache.
https://docs.djangoproject.com/en/1.10/howto/deployment/wsgi/modwsgi/
Cd to sites-available : etc/apache/sites-available
Make a copy just in case of something
Basic configuration in 000-default.config

```
<Directory /path/to/mysite.com/static>
Require all granted
</Directory>

<Directory /path/to/mysite.com/media>
Require all granted
</Directory>

WSGIScriptAlias / /path/to/mysite.com/mysite/wsgi.py

<Directory /path/to/mysite.com/mysite>
<Files wsgi.py>
Require all granted
</Files>
</Directory>
```

Log In to postgres on Server

Debian based systems like Ubuntu :

Connect/login as root -

```
user@user-pc:~$ sudo -i -u postgres
postgres@user-pc:~$ psql
psql (9.3.5, server 9.3.6)
Type "help" for help.
```

Connect to a database

\connect dbname