

IP & Hostname

Hostname: ec2-54-93-187-58.eu-central-1.compute.amazonaws.com

IP Address: 54.93.187.58

Application URL: http://ec2-54-93-187-58.eu-central-1.compute.amazonaws.com/

Amazon Lightsail Set Up:

- Create a new AWS account.
- Go to [Amazon Lightsail](#) website to create Instance.
- After create an instance Click the 'Download' to download your private key.
- Click the 'Networking' tab and find the 'Add another' at the bottom. Add port 123 and 2200.

Linux Configuration:

- Move your file .pem key into ssh folder
- In the terminal type to make our key secure:
- `$ chmod 600 ~/.ssh/filename.pem`
- Log into the server as the user ubuntu with our key. From the terminal type :
- `$ ssh -i ~/.ssh/filename.pem ubuntu@54.93.187.58`
- Now you will see the command line change to `root@[ip-your-private-ip]:$`
- Switch to the root user by typing:
- `sudo su -`
- Create a user called grader. From the command line type:
- `$ sudo adduser grader.`
- Create a file to give the user grader superuser privileges. By type this command line:
- `$ sudo nano /etc/sudoers.d/grader.`
- Then type:
- `grader ALL=(ALL:ALL)`
- Configuring a Linux server will update its package list, upgrade the current packages, and install new updates type these command lines:
`$ sudo apt-get update`
`$ sudo apt-get upgrade`
`$ sudo apt-get dist-upgrade`
- We will install a useful tool allow us to see the users on this server by typing this command:
`$ sudo apt-get install finger`
- Now we must create an SSH Key for our new user grader. From a new terminal run the command:
- `$ ssh-keygen -f ~/.ssh/udacity.rsa`
- We need to copy the public key using the command:

- `$ cat ~/.ssh/udacity.rsa.pub.`
- Go back to the first terminal move to grader's folder by:
- `$ cd /home/grader`
- Create `.ssh` directory:
- `$ mkdir .ssh`
- Create a file to store the public key with the command:
- `$ touch .ssh/authorized_keys`
- Edit the key by typing:
- `$ nano .ssh/authorized_keys`
- Then paste in the public key.
- Now will change the permissions of the file and its folder by typing:
- `$ sudo chmod 700 /home/grader/.ssh`
- `$ sudo chmod 644 /home/grader/.ssh/authorized_keys`
- Change the owner of the `.ssh` from root to grader by using the command:
- `$ sudo chown -R grader:grader /home/grader/.ssh`
- Finally we will restart its service with:
- `$ sudo service ssh restart`
- Disconnect from the server by typing: `$ ~.`
- Log back through port 2200:
- `$ ssh -i ~/.ssh/udacity_key.rsa -p 2200 grader@54.93.187.58`
- Disable ssh login for root user by running:
- `$ sudo nano /etc/ssh/sshd_config.`
- Change `PermitRootLogin` to `no`.
- Find Port 22 and change it to Port 2200.
- Change `PermitRootLogin` to `no`.
- Restart ssh:
- `$ sudo service ssh restart`
- Now we need to configure the firewall using these commands:
- `$ sudo ufw allow 2200/tcp`
- `$ sudo ufw allow 80/tcp`
- `$ sudo ufw allow 123/udp`
- `$ sudo ufw enable`

• Application Deployment

- First will install these software:
- `$ sudo apt-get install apache2`
- `$ sudo apt-get install libapache2-mod-wsgi python-dev`
- `$ sudo apt-get install git`
-
- Enable `mod_wsgi` with the command :
- `$ sudo a2enmod wsgi`
- Restart Apache using:

- `$ sudo service apache2 restart.`
- Input the public IP address in the web browser then you will see the Apache2 Ubuntu Default Page

- Now we have to create directory for our catalog application and make the user grader the owner.

- `$ cd /var/www`
- `$ sudo mkdir catalog`
- `$ sudo chown -R grader:grader catalog`
- `$ cd catalog`
- Now we clone the project from Github:
- `$ git clone https://github.com/Alwagdani/item-catalog.git catalog`

-

- Create a `.wsgi` file:
- `$ sudo nano catalog.wsgi`
- Add the following into this file:
- `import sys`
- `import logging`
- `logging.basicConfig(stream=sys.stderr)`
- `sys.path.insert(0, "/var/www/catalog/")`
- `from catalog import app as application`
- `application.secret_key = 'super_secret_key'`

- Rename `project.py`, [your catalog application folder] to `__init__.py` by typing:
- `$ mv project.py __init__.py`

- Now we will create our virtual environment:

- `$ sudo pip install virtualenv`
- `$ sudo virtualenv venv`
- `$ source venv/bin/activate`
- `$ sudo chmod -R 777 venv`

- You should see a `(venv)` appears before your username in the command line.

- We will install the Flask and other packages:

- `$ sudo apt-get install python-pip`
- `$ sudo pip install httplib2`
- `$ sudo pip install requests`
- `$ sudo pip install --upgrade oauth2client`
- `$ sudo pip install sqlalchemy`
- `$ sudo pip install flask`
- `$ sudo apt-get install libpq-dev`
- `$ sudo pip install psycopg2`

- change json path in `__init__.py` to be like this:

- `/var/www/catalog/catalog/client_secrets.json`

- Now we need to configure and enable the virtual host

- `$ sudo nano /etc/apache2/sites-available/catalog.conf`

- Paste the following code and save:

```
<VirtualHost *:80>
    ServerName 54.93.187.58
    ServerAlias ec2-54-93-187-58.eucentral-1.compute.amazonaws.com/
    ServerAdmin admin@54.93.187.58
    WSGIDaemonProcess catalog python-path=/var/www/catalog:/var/www/catalog/venv/lib/python2.7/
site-packages
    WSGIProcessGroup catalog
    WSGIScriptAlias / /var/www/catalog/catalog.wsgi
    <Directory /var/www/catalog/catalog/>
        Order allow,deny
        Allow from all
    </Directory>
    Alias /static /var/www/catalog/catalog/static
    <Directory /var/www/catalog/catalog/static/>
        Order allow,deny
        Allow from all
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/error.log
    LogLevel warn
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

- Now will set up the database

- `$ sudo apt-get install libpq-dev python-dev`

- `$ sudo apt-get install postgresql postgresql-contrib`

- `$ sudo su - postgres`

- You will see the username changed again in command line, and type `$ psql`

```
postgres@ip-172-26-6-215:~$ psql
```

- Now we will create a user to create and set up the database with database called `catalog` with user called `catalog`:

- `$ CREATE USER catalog WITH PASSWORD '12345';`

- `$ ALTER USER catalog CREATEDB;`

- `$ CREATE DATABASE catalog WITH OWNER catalog;`

- To connect to database use this command line :

- `$ \c catalog`

- `$ REVOKE ALL ON SCHEMA public FROM public;`

- \$ GRANT ALL ON SCHEMA public TO catalog;
- Use sudo nano command to change all engine to engine = create_engine('postgresql://catalog:12345@localhost/catalog')
- we will do it for __init__.py, and database_setup.py
- Restart your apache server
- \$ sudo service apache2 restart
- and now put your IP address in the browser.
- **In Google Developers Console :**
- **in Authorized JavaScript origins add:**

http://ec2-54-93-187-58.eu-central-1.compute.amazonaws.com	
------------------------------------------------------------	--

http://54.93.187.58.xip.io	
----------------------------	--

http://54.93.187.58	
---------------------	--

- **In Authorized redirect URIs add:**

http://54.93.187.58.xip.io/	
-----------------------------	--

http://ec2-54-93-187-58.eu-central-1.compute.amazonaws.com/login	
------------------------------------------------------------------	--

http://ec2-54-93-187-58.eu-central-1.compute.amazonaws.com/gconnect	
---------------------------------------------------------------------	--

http://ec2-54-93-187-58.eu-central-1.compute.amazonaws.com/	
-------------------------------------------------------------	--

- **In Authorized domains add:**

ec2-54-93-187-58.eu-central-1.compute.amazonaws.com	
-----------------------------------------------------	--

xip.io	
--------	--

amazonaws.com	
---------------	--

Reference:

Thank you so much for those who posted a guide step-by-step to do the project on their Github:

<https://github.com/chuanqin3/udacity-linux-configuration>

<https://github.com/mulligan121/Udacity-Linux-Configuration>