**How to set up AWS Instance for this project?**

# Creating an EC2 Instance on AWS for Model Training & API Deployment :

Step 1 : Select the EC2 option from the “Services” dropdown on your aws dashboard.

Step 2 : On the next screen of EC2 Dashboard, select the “Launch Instance” option.

Step 3 : In this project, we’ll use Ubuntu Server 16.04 based machine, so select that from the list of choices for machine image.

Step 4 : Now select the instance type - t2.micro (It comes under the Free Tier Label). Also, this is the default choice. So, if it’s already locked in, don’t change anything.

Step 5. In “Configure Instance details”, enable Auto Assign IP - to open access to the machine through the internet.

Step 6. In next step of adding storage, just use the Default Settings and continue.

Step 7. In the next step of adding tags, again use the default settings(or skip this step) and continue.

Step 8. Configure the security groups as shown below to have 2 rules - SSH & HTTP. Initially it has only one rule ie, SSH.

Step 9. Save the .pem key to access the EC2 Machine and then launch.

Step 10. Set up an elastic IP for the instance. Elastic IP - remains the same (whereas the ip which comes associated by default with the instance changes) when you restart the instance, thus giving you better control over the instance.

Step 11. Associate the elastic IP Address created in the last step with your EC2 instance.

**How to connect to EC2 Instance using Putty & Puttygen ?**

Step 1. Convert key from .pem to .ppk. To do this you need to start “Puttygen” and press the Load button in “Puttygen”. You need to load your .pem key file which you downloaded from aws while creating the EC2 Instance.

Step 2. Then you need to save the private key by pressing the “Save Private Key” Button.

Step 3. You need to save it without the paraphrase. So ignore the warning. Just hit “Yes”.

Step 4. Setup Putty by entering the hostname (i.e the url of your instance). You can extract the hostname (or url) of the instance by copying the url from Public DNS from aws instance page.

Step 5. Set the .ppk key in Putty in the field by accessing Connection>>SSH>>Auth>>Browse(collect your saved file). Then click Open.

Step 6. Open the Connection to check if it’s working fine or not.

**How to set up Filezilla for File Transfer?**

Step 1. Add new site in site manager.

Step 2. Add ‘Amazon AWS key (.pem)’ in Edit->Settings->SFTP

Step 3. Connect to the new site added using the site manager. Now you are ready to transfer your files.

**How to configure Apache server on EC2?**

Step 1. Install apache server using the following command : sudo apt-get install apache2

Step 2. Install apache webserver gateway interface using the following command : sudo apt-get install libapache2-mod-wsgi

Visit https://.com to see if everything is working fine. If you installed it correctly you will see a apache welcome page.

**How to set up the Flask app on EC2 Instance?**

Step 1. Open the terminal & create the following directory

> /var/www/FlaskApplications

Step 2. Now create the following directory

> /var/www/FlaskApplications/SampleApp

Step 3. Change the permissions of both the above directories using the following commands :

1. sudo chown -R ubuntu:ubuntu /path/to/directory

2. sudo chmod -R 777 /path/to/directory

NOTE: “/path/to/directory” obviously refers to the path of directory/file of which you want to change the permissions. We change the permissions so that we and the apache server (and in return the internet) are able to write at that location.

Step 4. Change the hostname in SampleApp.conf to match your EC2 Instance Hostname (or url)

Step 5. Place the SampleApp.conf file at

> /etc/apache2/sites-available/SampleApp.conf

Step 6. Place the .wsgi file at

> /var/www/FlaskApplications/

Step 7. Place the demo.py file at

> /var/www/FlaskApplications/SampleApp/api

Step 8. Execute the following command to initialize the apache server and the api

1. sudo a2enmod wsgi

2. sudo apachectl restart

3. sudo a2ensite sampleApp

Step 9. Execute the following command to start the server:

1. sudo service apache2 reload

2. sudo /etc/init.d/apache2 reload

NOW THE SETUP IS COMPLETE ^\_^ CHEERS!