Sprint 4: Hosting your project on AWS

Directions:

This week is the final week of the first project. The goal of this week is to host your project on a remote server so other people can use your app. To accomplish this one person in the group will need to make an AWS account and you should then work together to launch an instance, set-up a web-server, set up the database on the instance, secure the instance, and make it available to the outside world.

The second goal for this week is for you to clean up your existing work and fix any issues from the previous sprints.

As always, the lectures are available on the course website.

Remember for this week we want you to use the free tier when setting up a server on EC2 the name of the free tier instances are "Micro". If you have any concerns about this please ask us or come to see us. No one should be charged any money for this.

AWS Free:
https://aws.amazon.com/free

EC2:
Getting Started with EC2
Accessing Instances Linux

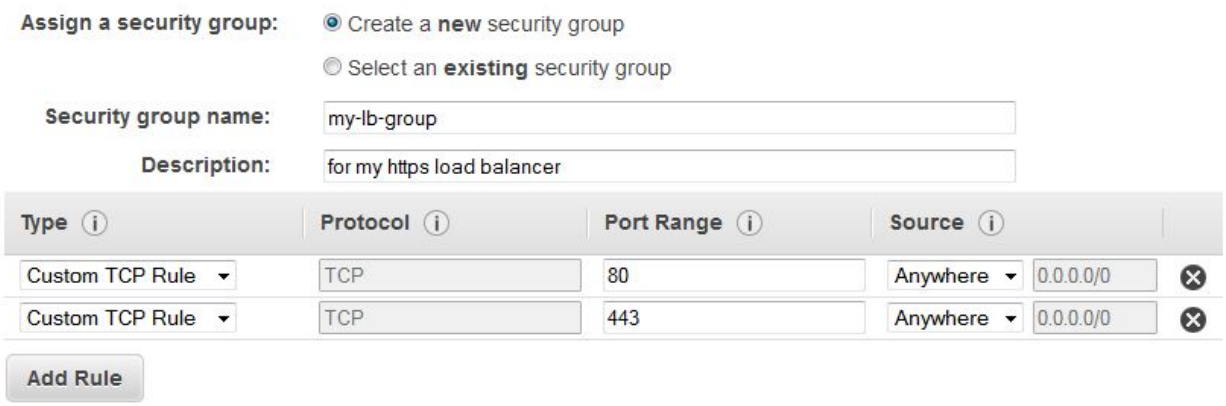
EC2 Instance Setup: *(33 points)*

Your group must launch a Micro Tier (free) EC2 instance.

When setting up the instance, make sure to choose the "Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-66506c1c" on the first page.

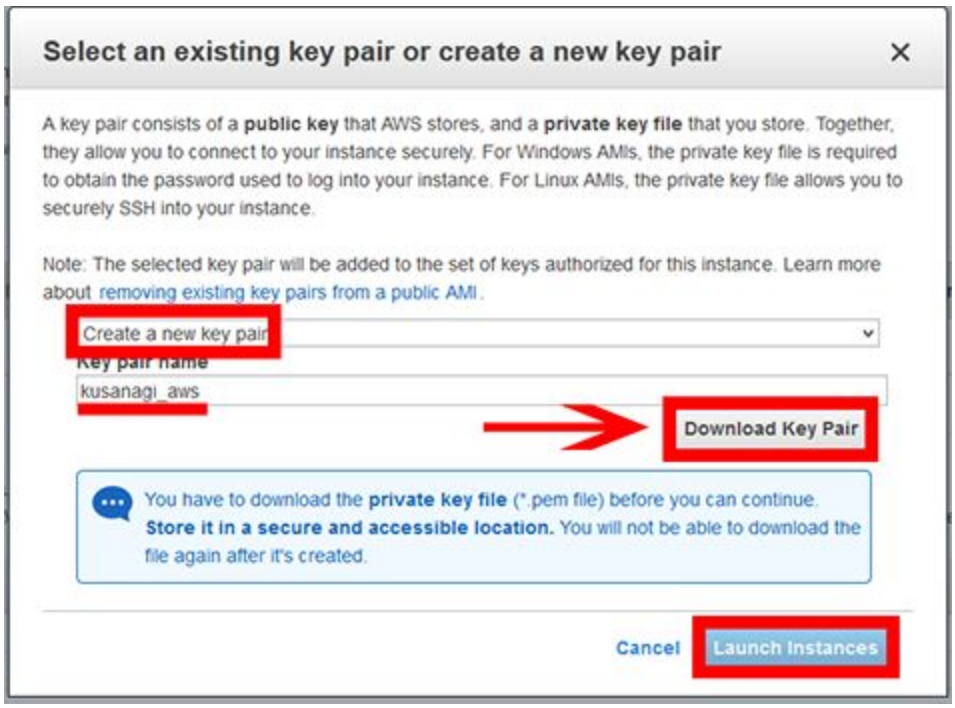
Security Groups:

When you are launching your instance, you should configure the security groups to allow inbound traffic to your instance on port 22 (SSH), 80 (HTTP), and 443 (HTTPS).



Creating a new key pair:

On the final step, you will see the following:



Name the key pair and download it. **DO NOT LOSE THIS FILE.** You and your team will not be able to access the instance without it. Be sure to give a copy of the key file to any of your team members who need to access the instance.

Use the terminal on the virtual machine we provided you (or on your Mac if you have one) to ssh into the instance you launched.

Ex: *ssh -i /path/to/key.pem ubuntu@<your-public-domain>*

If you get an error saying that your key is insecure, you will need to run *chmod 400 <yourkey>.pem* on it.

NGINX SETUP: *(33 points)*

Nginx is the web server that will help make your project safely available to the outside world.

At the very least the default NGINX welcome page should be accessible from the link to your instance.

After this if NGINX is hosting your project this page will go away, but we will still be able to see that you are hosting via NGINX.

Installing nginx after connecting to your instance:

```
sudo apt update
sudo apt install python-software-properties
nginx=stable
sudo add-apt-repository ppa:nginx/$nginx
sudo apt update
sudo apt install nginx
```

Your web app must be publicly accessible from port 80 or port 443 (if you have set up HTTPS). You will also need to allow access to these ports for your instance from AWS.

Setting up your Flask app with NGINX and Gunicorn: *(33 points)*

https://www.digitalocean.com/community/tutorials/how-to-serve-flask-applications-with-uwsgi-and-nginx-on-ubuntu-16-04

You should also remember that these are not the only available resources on the topic. Before coming to office hours or emailing us for help it is your responsibility as self-interested learners to search. Google and stack-overflow are your friends and the questions you will have are easily answerable one google away.

Set up your project to be accessible to the world: NGINX should be hosting your project which should now be accessible publicly on port 80.

Everything you've set up on your own laptop (locally), you will have to set-up on the instance, including Flask, PyMyMSQL, MYSQL, etc. You will need to copy your code to the server and recreate your database on the server.

EXTRA CREDIT - SET UP HTTPS (10 points)

HTTPS is a protocol that encrypts data traveling to and from your website. If someone is eavesdropping on your connection they cannot steal information because they won't be able to decrypt it.

FINAL STEPS:

Fix your errors from previous weeks:

Now that you have finished setting up the project on a remote server, you should fix any errors you have in the code from previous sprints.

Earn back points you may have lost from the previous sprints.

Turning in your project:

We will only be grading the code in the master branch of your repository. We will not be looking at any other branches.

Your github repo for the group project must include a README.md in the root that includes the following:

- Team name
- Name and net ids of each group member
- The public url to your app hosted on AWS
- A brief description of your project and its dependencies (the frameworks/libraries used)
- Project map and description of each file (example below)

```
app.py          - contains all app routes and starts server
README.md       - this file
templates/      - contains html files rendered to browser
     index.html      - home page
static/         - contains static css, js, and image files
     css/            - contains css files
          style.css       - all css styling code
     js/             - contains javascript files
          requests.js     - js making requests to back end
     img/            - contains all images used
          logo.jpg        - logo image
```

- Instructions for how to run the application

Project 1 Survey:

Every member of the group must fill out the following survey to receive a grade for this project. Your feedback will help us improve your experience going forward.

https://goo.gl/forms/XAMmZGLzzl6kvdvm2