Documentation

Project0 SoftBear-Eng

Hope Anderson

Bryant Hall

Blake Gerard

Cory White

Tim Gehrsitz

Python VM setup - Cory

1. Go to GCP and go to the compute engine
2. Go to vm instances and create a new machine
3. For our purpose I just used a g1-small, probably could go small if desired
4. Allow https and https traffic and click create
5. Once the instance is created click on it and scroll down to network interfaces
6. Click the view details button on the far right of the default network row
7. Here on the left side of the page click firewall rules
8. At the top of the page click create firewall rule
9. Give it a name and a description if desired.
10. In the target tags input flask
11. In the source ip ranges input 0.0.0.0/0
12. Next select tcp check box in the protocols and ports and in the textbox input 5000
13. Then select create
14. Once that is finished creating go back to your VM instance list and open up the ssh to the newly created VM
15. Before we begin we will need to install and update a few things
16. Run the following commands
    1. sudo apt-get update
    2. make sure python is install by running python –version if not run sudo apt-get python
    3. now we will make a directory for our project run mkdir rng-proj
    4. run cd rng-proj
    5. we can download and create a python virtual environment so all the dependencies we download are contained in this directory so if we use git anyone will have the necessary files, but we will not google how to do that if desired should be something along the lines of pip install virtualenv and then doing the command virtualenv env but we will not do that
    6. we will now want to install pip so run sudo apt-get install python-pip and input y when it asks for confirmation
    7. when finished run sudo pip install flask
    8. now we want to make the flask python file to run the “website”
17. using your text editor of choice make a file, we will name it for simplicity sake server.py
18. in this file paste this text

import random

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route(‘/’)

Def index():

rn= random.randint(1,1000000)

return str(rn)

if \_\_name\_\_ == “\_\_main\_\_”:

app.run(host=’0.0.0.0’,port=80)

19.Make sure if you copy paste that all the code is formatted correctly and that the string and character ‘ “ etc are all correct

20.save and close the file

21. run the command sudo python server.py if you installed and set everything up correctly it should say something about

\* Serving Flask app "server" (lazy loading)

\* Environment: production WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

\*Debug mode: off \* Running on <http://0.0.0.0:80/> (Press CTRL+C to quit)

If this shows your server should be running.

22. to connect to the server you will use your vms external ip but we need to use http to connect not https so the url for example would be <http://11.111.11.11/> if everything was done properly you should see your random number and it should change everytime the page is refreshed or connected to. Anyone going to that url should be able to connect and see the random number

Side note- the fire wall rule we setup may have not been necessary.

# Chapter 2 – Bryant

Java App-Engine

1. Load up a new project in GCP
2. Go to App Engine Dashboard
3. Click start tutorial and select Java.
4. Run through the tutorial and deploy the tutorial application
5. Open Shell (ensure you are in directory appengine-try-java) and launch code editor (pencil icon at the top of the cloud shell console) for the tutorial app
6. Enter the DemoServlet.java file by following the directory path of appengine-try-java/src/main/java/myapp/demoserverlet.java
7. Within the doGet() function type:

“int randomNumber = (int)(Math.random() \* 1000000);”

8. Remove:

“resp.getWriter().println("{ \"value\": \"World\"}");”

1. Replace with:

PrintWriter out = resp.getWriter();

out.printf("{ \"value\": \"%d\"}", randomNumber);

1. Navigate to index.html in the directory path appengine-try-java/src/main/webapp/index.html and delete the line

$('#result').html("Hello, " + data.name);

and replace with

$('#result').html(data.value);

1. Return to Bash shell and type:

“gcloud config set project \ **<YOUR-PROJECT-ID>”**

(your project-ID can be found at GCP Home dashboard)

“mvn appengine:deploy”

1. Here is my working link: <https://javaproject0-251917.appspot.com/>

Ls