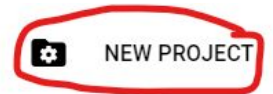


Deployment Instructions

Java App Engine


1. Create a new project in Google Cloud Platform

Select a project




2. Name the project as you please and then click on "CREATE"


New Project

 You have 17 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name *
java-app-engine-test 

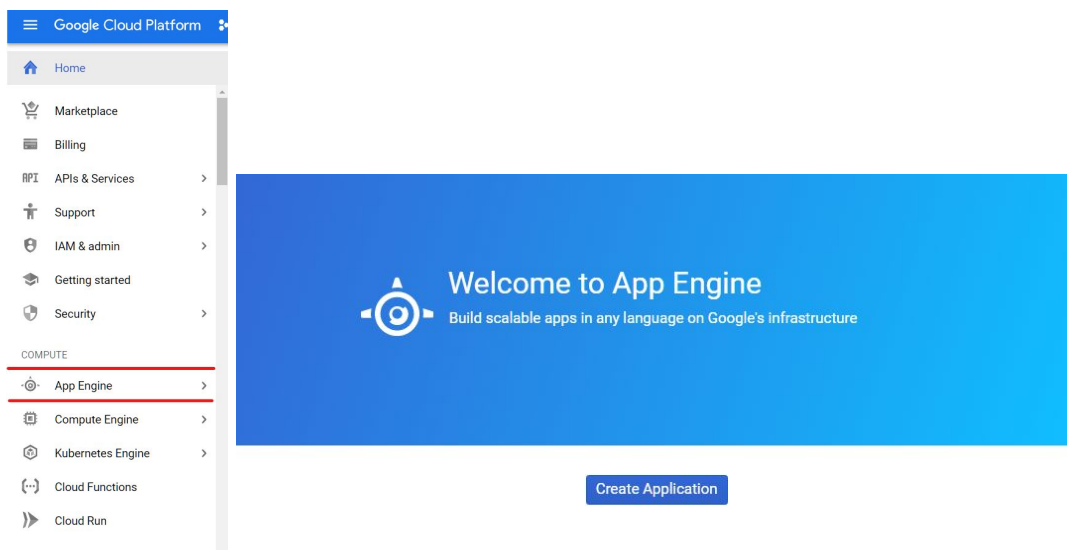
Project ID: poised-cortex-254518. It cannot be changed later. [EDIT](#)

Location *
 No organization [BROWSE](#)

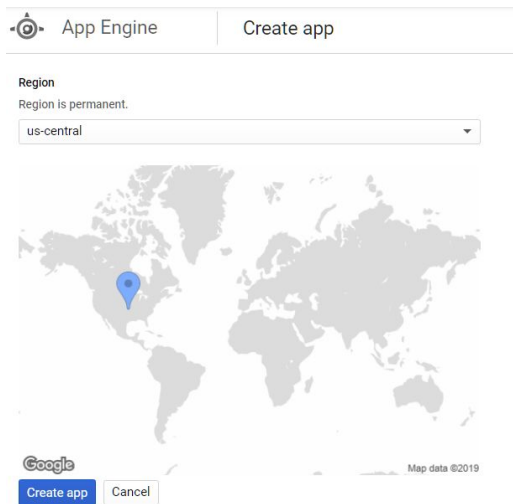
Parent organization or folder

[CREATE](#) [CANCEL](#)

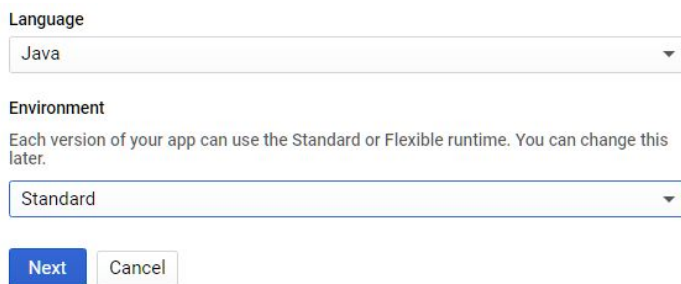
3. Navigate to the App Engine page and click on "Create Application"



4. Select an appropriate server location and click on "Create app"



5. Select "Java" as the language and select "Standard" as the environment



6. Open Google Cloud Shell



7. Clone this repository with the command "git clone <https://github.com/Andy-Vu-Viz/RandomNumberGen-Servlets/>"

```
gcloudshell:~ (poised-cortex-254518) $ git clone https://github.com/Andy-Vu-Viz/RandomNumberGen-Servlets/
Cloning into 'RandomNumberGen-Servlets'...
remote: Enumerating objects: 76, done.
remote: Counting objects: 100% (76/76), done.
remote: Compressing objects: 100% (62/62), done.
remote: Total 1049 (delta 22), reused 5 (delta 1), pack-reused 973
Receiving objects: 100% (1049/1049), 6.76 MiB | 0 bytes/s, done.
Resolving deltas: 100% (136/136), done.
```

8. Also clone this repository with the command “git clone <https://github.com/GoogleCloudPlatform/getting-started-java>”

```
cloudshell:~ (poised-cortex-254518)$ git clone https://github.com/GoogleCloudPlatform/getting-started-java
Cloning into 'getting-started-java'...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 7714 (delta 9), reused 11 (delta 5), pack-reused 7687
Receiving objects: 100% (7714/7714), 50.80 MiB | 50.27 MiB/s, done.
Resolving deltas: 100% (4191/4191), done.
```

9. Use the command “cp RandomNumberGen-Servlets/java-appengine-randomnumbergenerator-master/src/ getting-started-java/appengine-standard-java8/helloworld -r”

```
(poised-cortex-254518)$ cp RandomNumberGen-Servlets/java-appengine-randomnumbergenerator-master/src/ getting-started-java/appengine-standard-java8/helloworld -r
```

10. Navigate into the “helloworld” directory using “cd getting-started-java/appengine-standard-java8/helloworld”

```
cloudshell:~ (poised-cortex-254518)$ cd getting-started-java/appengine-standard-java8/helloworld
cloudshell:~/getting-started-java/appengine-standard-java8/helloworld (poised-cortex-254518)$ mv
```

11. Execute command "mvn appengine:deploy" and wait for it to complete

```
cloudshell:~/getting-started-java/appengine-standard-java8/helloworld (poised-cortex-254518)$ mvn appengine:deploy
[INFO] Scanning for projects...
[INFO] -----< com.example.appengine-j8:helloworld >-----
[INFO] Building helloworld 1.0-SNAPSHOT
[INFO] -----[ war ]-----
```

12. Navigate to the link shown on the dashboard page of App Engine and click on “Random Number Generator” to receive a randomly generated number

[Random Number Generator](#)

105220

Java VM

1. Download the WAR file named "Servlets.war" from the repository
<https://github.com/Andy-Vu-Viz/RandomNumberGen-Servlets/>

The screenshot shows the GitHub repository page for 'Andy-Vu-Viz / RandomNumberGen-Servlets'. The repository has 44 commits, 2 branches, 0 releases, and 3 contributors. The file list includes 'RandomNumberFisk', 'RandomNumberPythonAE', 'RandomNumberTomcat-master/Servlets', 'TestResults', 'helloworld', 'java-appengine-randomnumbergenerator-mast...', 'README.md', 'Servlets.war', and 'testscript.py'. The 'Servlets.war' file is highlighted with a green box. Below the file list, the details for 'Servlets.war' are shown, including the commit hash 'ec19827' and the file size '2.14 KB'. The 'Download' button is highlighted with a green box.

Andy-Vu-Viz / RandomNumberGen-Servlets

44 commits 2 branches 0 releases 3 contributors

Branch: master New pull request Create new file Upload files Find File Clone or download

Andy-Vu-Viz starting/jumping off point for java appengine Latest commit a792da1 5 days ago

File	Commit	Time
RandomNumberFisk	Update supervisor_config.txt	16 days ago
RandomNumberPythonAE	add python app engine	17 days ago
RandomNumberTomcat-master/Servlets	Java Servlets	19 days ago
TestResults	add test log	16 days ago
helloworld	starting/jumping off point for java appengine	5 days ago
java-appengine-randomnumbergenerator-mast...	Java Servlets	19 days ago
README.md	Update README.md	16 days ago
Servlets.war	Java Tomcat Servlet WAR File	16 days ago
testscript.py	fix test script and add results	16 days ago

Servlets.war

ec19827 16 days ago

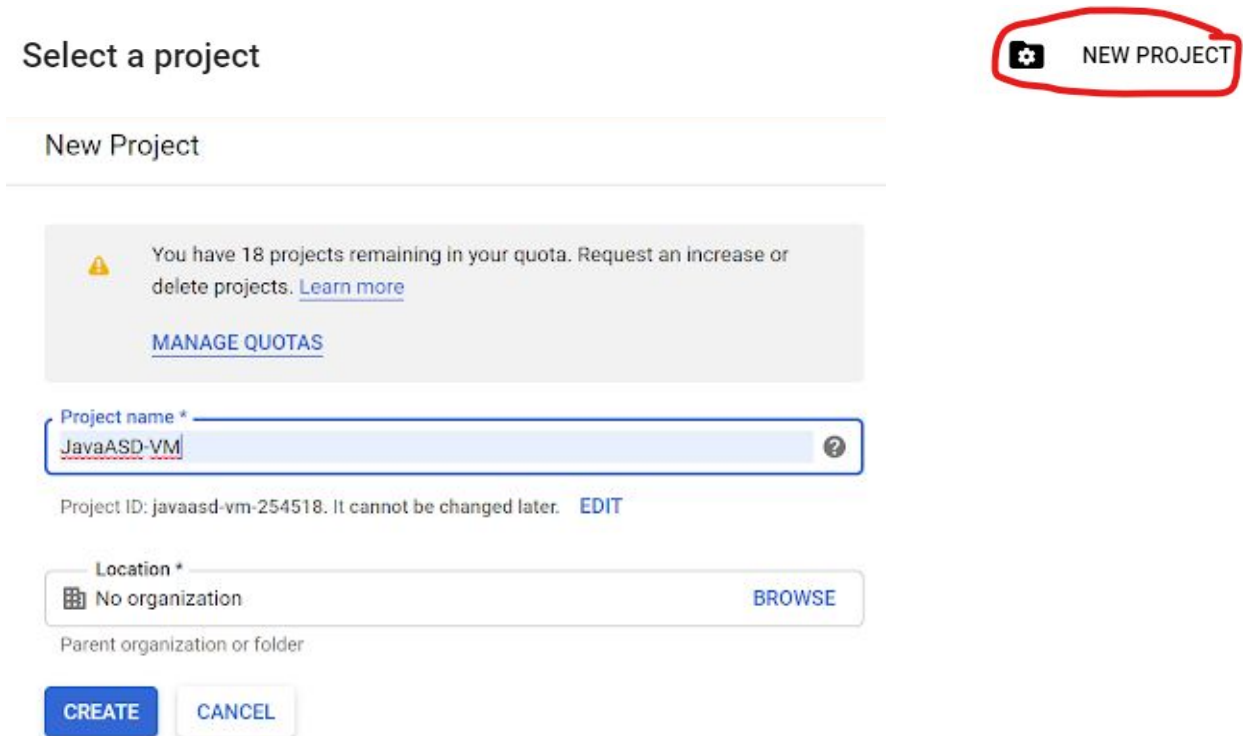
2.14 KB Download history

View raw


© 2019 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

2. Create a new project in Google Cloud Platform

Select a project




New Project

 You have 18 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name *
JavaASD-VM

Project ID: javaasd-vm-254518. It cannot be changed later. [EDIT](#)

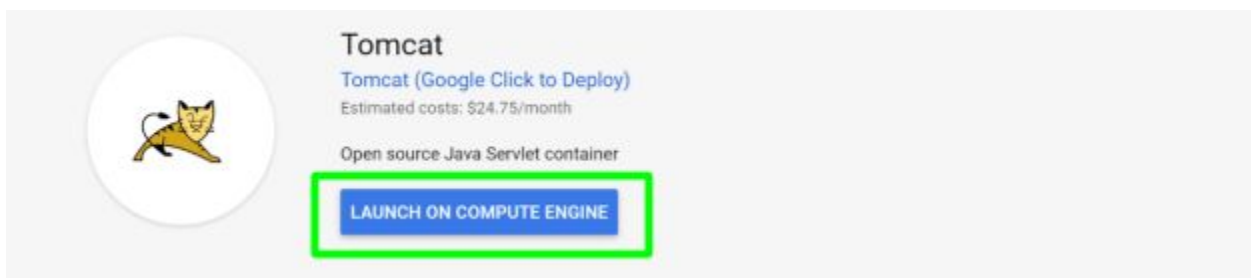
Location *
 No organization [BROWSE](#)


Parent organization or folder

[CREATE](#) [CANCEL](#)

3. Navigate to the link

"<https://console.cloud.google.com/marketplace/details/click-to-deploy-images/tomcat>"





Tomcat
Tomcat (Google Click to Deploy)
Estimated costs: \$24.75/month
Open source Java Servlet container

[LAUNCH ON COMPUTE ENGINE](#)

Runs on
Google Compute Engine

Type
[Virtual machines](#)
Single VM

Overview

Tomcat is an open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies. It can be used to simplify developing web applications and web services.

[Learn more](#) 

4. Click on "LAUNCH ON COMPUTE ENGINE"

5. Select the project you created

6. Leave the settings as default except for changing the Zone to an appropriate location
7. Scroll to the bottom of the page and click "Deploy"

← New Tomcat deployment

Deployment name

tomcat-1

Zone ⓘ

us-east1-b

Machine type ⓘ

1 vCPU

3.75 GB memory

Customize

Boot Disk

Boot disk type ⓘ

Standard Persistent Disk

Boot disk size in GB ⓘ

10

Networking

Network interfaces

default default (10.142.0.0/20)

+ Add network interface

Firewall ⓘ

Add tags and firewall rules to allow specific network traffic from the Internet

☒ Allow HTTP traffic from the Internet

Source IP ranges for HTTP traffic ⓘ

0.0.0.0/0, 192.169.0.2/24

☐ Allow HTTPS traffic from the Internet

Source IP ranges for HTTPS traffic ⓘ

0.0.0.0/0, 192.169.0.2/24


Stackdriver

Monitoring and management for services, containers, applications, and infrastructure

☐ Enable Stackdriver Logging ⓘ

☐ Enable Stackdriver Monitoring ⓘ

Deploy



Tomcat overview

Solution provided by Google Click to Deploy

\$24.75 per month estimated

Effective hourly rate \$0.034 (730 hours per month)

Details

Software

Operating System	Debian (9.11)
Software	Apache (2.4.25)
	Google-Fluentd (1.6.17)
	OpenJDK (1.8.0)
	Stackdriver-Agent (5.5.2)
	Tomcat (8.5.14.0)

Terms of Service

The software or service you are about to use is not a Google product. By deploying the software or accessing the service you are agreeing to comply with the [GCP Marketplace terms of service](#) and the terms of any third party software licenses related to the software or service. Please review these licenses carefully for details about any obligations you may have related to the software or services. To the limited extent an open source software license related to the software or service expressly supersedes the GCP Marketplace Terms of Service, that open source software license governs your use of that software or service.

By using this product, you understand that certain account and usage information may be shared with Google Click to Deploy for the purposes of sales attribution, performance analysis, and support. ⓘ

Google is providing this software or service 'as-is' and will not perform any ongoing maintenance. Ongoing upgrades and maintenance are your responsibility.

8. Wait for the deployment operation to finish

The screenshot shows the Google Cloud Platform Deployment Manager interface. The left sidebar has 'Deployments' and 'Type registry'. The main area shows the deployment 'tomcat-1' with a status of 'tomcat-1 is being deployed'. Below this, a tree view shows the deployment steps: 'tomcat-1-jinja', 'tomcat-vm-templ', 'tomcat-1-vm', 'generated-password-0', 'software-status', 'tomcat-1-config', 'tomcat-1-software', and 'tomcat-1-tcp-80'. The right sidebar shows the 'Tomcat' solution provided by Google Click to Deploy, with a table of configuration details.

Property	Value
Site address	Pending
Admin user	tomcat
Admin password (Temporary)	Pending
Instance	Pending
Instance zone	Pending
Instance machine type	Pending

More about the software

Get started with Tomcat

You will be able to use Tomcat after the deployment is completed.

Documentation

[Tomcat Documentation](#)

Support

Google does not offer support for this solution. However, community support is available on Stack Overflow. Additional support is available on community forums.

Template properties

[More](#)

9. You may follow the "Suggested next steps" if you choose to 10. Click on the "Visit the site" button

The screenshot shows the Google Cloud Platform Deployment Manager interface. The left sidebar has 'Deployments' and 'Type registry'. The main area shows the deployment 'tomcat-1' with a status of 'tomcat-1 has been deployed'. Below this, a tree view shows the deployment steps: 'tomcat-1-jinja', 'tomcat-vm-templ', 'tomcat-1-vm', 'generated-password-0', 'software-status', 'tomcat-1-config', 'tomcat-1-software', and 'tomcat-1-tcp-80'. The right sidebar shows the 'Tomcat' solution provided by Google Click to Deploy, with a table of configuration details.

Property	Value
Site address	http://184.194.141.14/
Admin user	tomcat
Admin password (Temporary)	abK75h4re
Instance	tomcat-1-vm
Instance zone	us-east1-b
Instance machine type	e1-standard1

More about the software

Get started with Tomcat

Visit the site SSH

Suggested next steps

- Open HTTPS traffic: This firewall rule is not enabled. To allow specific network traffic from the internet, create a firewall rule to open HTTPS traffic for target tag 'tomcat-1-deployment'. [Learn more](#)
- If you are using Google Cloud SDK, type the following command in the terminal:

```
gcloud --project=javaasdvms compute firewall-rules create "tomcat-1-tcp-443" --net
```
- Change the temporary password

11. On the webpage that just opened click on the hyperlink "manager webapp" and login using the information provided in the deployment manager

It works !

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

This is the default Tomcat home page. It can be found on the local filesystem at: `/var/lib/tomcat8/webapps/ROOT/index.html`

Tomcat8 veterans might be pleased to learn that this system instance of Tomcat is installed with `CATALINA_HOME` in `/usr/share/tomcat8` and `CATALINA_BASE` in `/var/lib/tomcat8`, following the

You might consider installing the following packages, if you haven't already done so:

tomcat8-docs: This package installs a web application that allows to browse the Tomcat 8 documentation locally. Once installed, you can access it by clicking [here](#).

tomcat8-examples: This package installs a web application that allows to access the Tomcat 8 Servlet and JSP examples. Once installed, you can access it by clicking [here](#).

tomcat8-admin: This package installs two web applications that can help managing this Tomcat instance. Once installed, you can access the [manager webapp](#) and the [host-manager webapp](#).

NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager webapp is restricted to users with role "admin-gui". Users are defined

12. Under the "Deploy" section of the webpage go to the subsection labeled "WAR file to deploy" and click on "Choose File" and choose the WAR file you downloaded and then click "Deploy"

The screenshot shows the Tomcat Web Application Manager interface. The "Deploy" section is expanded, and the "WAR file to deploy" subsection is highlighted with a green box. This subsection contains a "Choose File" button and a "Deploy" button. The "WAR file to deploy" subsection also includes a "Refresh WAR file to upload" button and a "Refresh WAR file" button. The "WAR file to deploy" subsection also includes a "Deploy" button.

Path	Version	Display Name	Running	BaseURL	Commands
/	None specified		True	8	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes
/docs	None specified	Tomcat Documentation	True	8	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes
/examples	None specified	Servlet and JSP Examples	True	8	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes
/documentation	None specified	Tomcat Host Manager Application	True	8	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes
/manager	None specified	Tomcat Manager Application	True	8	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes

Deploy

Deploy directory or WAR file located on server

Context Path (required):
WAR Configuration file URL:
Web or Directory URL:
Deploy

WAR file to deploy

Refresh WAR file to upload: Choose File Refresh WAR
Deploy

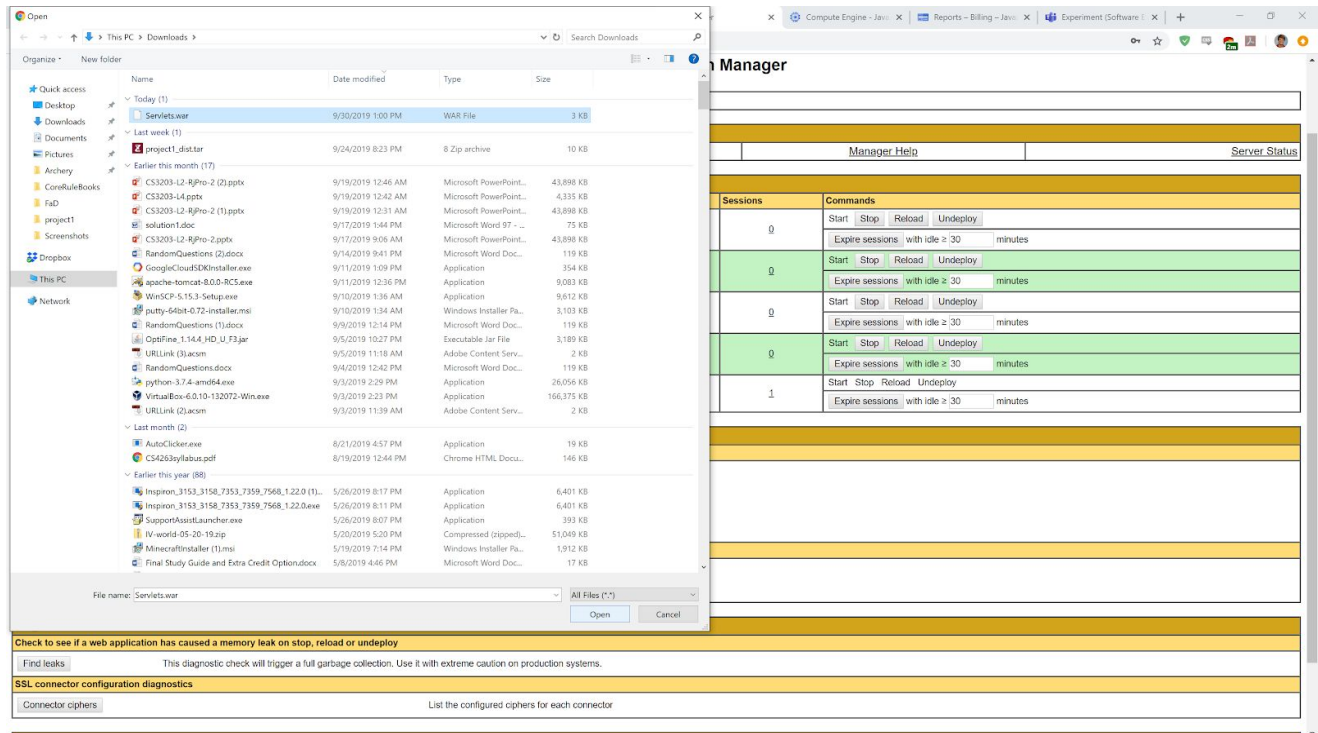
Diagnostic

Check to see if a web application has caused a memory leak on stop, reload or undeploy

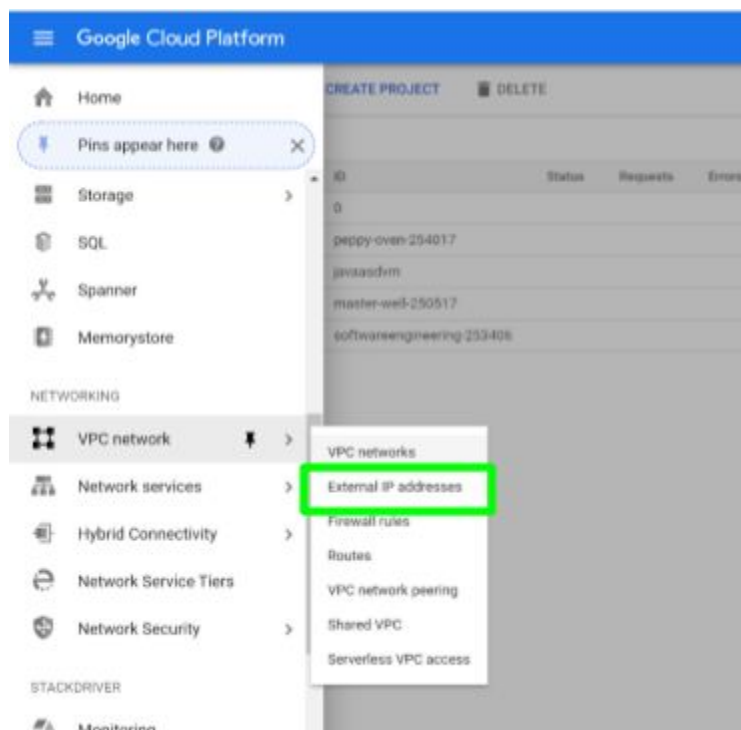
First tests: This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems.

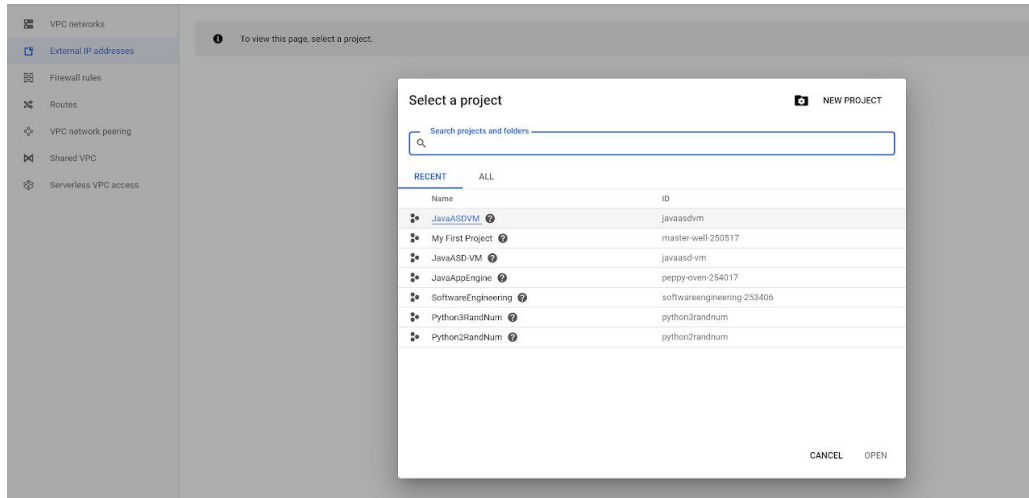
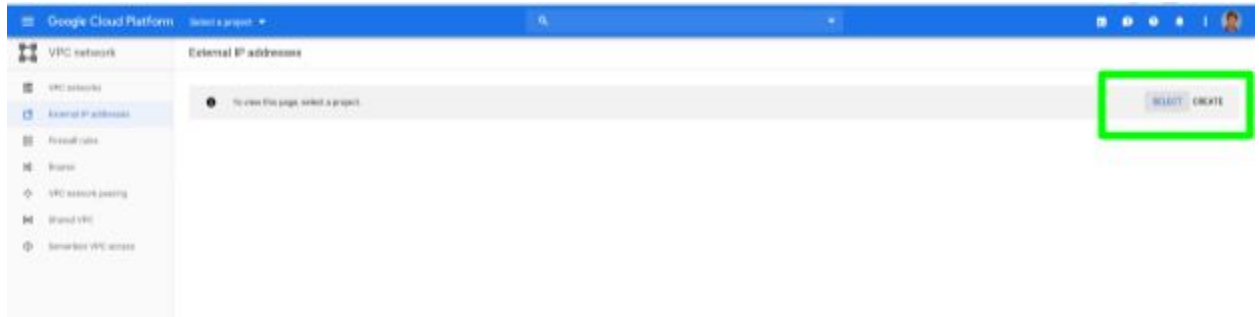
SSL, connector configuration diagnostics

Connector options: List the configured options for each connector



13. Setup a static IP for your virtual machine by going to the VPC Network page on GCP

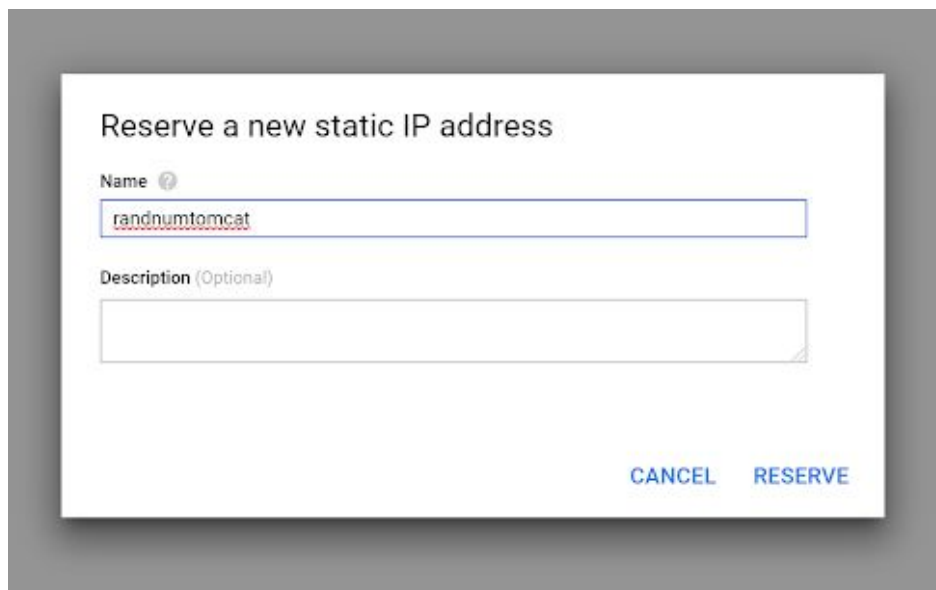




14. Under the "External IP addresses" find the IP address being used by the VM instance containing your Tomcat deployment



15. Switch "Ephemeral" to "Static" and reserve an IP with any name you find appropriate



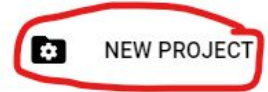
16. To access the random number generator navigate to "[The IP Address of the Server]/Servlets/RandomNumberGen"



Python App Engine

1. Create a new project in Google Cloud Platform

Select a project



2. Open the Cloud Shell



3. Clone our repository with the command "git clone <https://github.com/Andy-Vu-Viz/RandomNumberGen-Servlets/>"

```
@cloudshell:~/python-appengine-test (python-ae-test-2)$ git clone https://github.com/Andy-Vu-Viz/RandomNumberGen-Servlets/
Cloning into 'RandomNumberGen-Servlets'...
remote: Enumerating objects: 83, done.
remote: Counting objects: 100% (83/83), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 1056 (delta 23), reused 5 (delta 1), pack-reused 973
Receiving objects: 100% (1056/1056), 6.76 MiB | 0 bytes/s, done.
Resolving deltas: 100% (137/137), done.
@cloudshell:~/python-appengine-test (python-ae-test-2)$
```

4. Switch to the PythonRandomNumberAE directory with "cd RandomNumberGen-Servlets/RandomNumberPythonAE"

```
@cloudshell:~/python-appengine-test (python-ae-test-2)$ cd RandomNumberGen-Servlets/RandomNumberPythonAE
@cloudshell:~/python-appengine-test/RandomNumberGen-Servlets/RandomNumberPythonAE (python-ae-test-2)$
```

5. Install a virtual environment by running the command "virtualenv --python python3 venv"

```
@cloudshell:~/python-appengine-test/RandomNumberGen-Servlets/RandomNumberPythonAE (python-ae-test-2)$ virtualenv --python python3 venv
Running virtualenv with interpreter /usr/local/bin/python3
Already using interpreter /usr/local/bin/python3
Using base prefix '/usr/local'
New python executable in /home/.../python-appengine-test/RandomNumberGen-Servlets/RandomNumberPythonAE/venv/bin/python3
Also creating executable in /home/.../python-appengine-test/RandomNumberGen-Servlets/RandomNumberPythonAE/venv/bin/python
Installing setuptools, pip, wheel...
done.
@cloudshell:~/python-appengine-test/RandomNumberGen-Servlets/RandomNumberPythonAE (python-ae-test-2)$
```

6. Activate the virtual environment by running "source venv/bin/activate"

```
@cloudshell:~/python-appengine-test/RandomNumberGen-Servlets/RandomNumberPythonAE (python-ae-test-2)$ source venv/bin/activate
(venv) @cloudshell:~/python-appengine-test/RandomNumberGen-Servlets/RandomNumberPythonAE (python-ae-test-2)$
```

Now (venv) should precede [your username]@cloudshell:[path to RandomNumberPythonAE]

7. Install project dependencies with "pip install -r requirements.txt"

```
(venv) @cloudshell:~/python-appengine-test/RandomNumberGen-Servlets/RandomNumberPythonAE (python-ae-test-2) $ pip install -r requirements.txt
Collecting Flask==1.0.2 (from -r requirements.txt (line 1))
  Using cached https://files.pythonhosted.org/packages/7f/e7/08578774ed4536d3242b14dadb4696386634607af824ea997202cd0edb4b/Flask-1.0.2-py2.py3-none-any.whl
Collecting itsdangerous>=0.24 (from Flask==1.0.2->-r requirements.txt (line 1))
  Using cached https://files.pythonhosted.org/packages/76/ae/44b03b253d6fade317f32c24d100b3b35c2239807046a4c953c7b89fa49e/itsdangerous-1.1.0-py2.py3-none-any.whl
Collecting Jinja2>=2.10 (from Flask==1.0.2->-r requirements.txt (line 1))
  Using cached https://files.pythonhosted.org/packages/1d/e7/fd8b501e7a6dfe492a433deb7b9d833d39ca74916fa8bc63dd1a4947a671/Jinja2-2.10.1-py2.py3-none-any.whl
Collecting click>=5.1 (from Flask==1.0.2->-r requirements.txt (line 1))
  Using cached https://files.pythonhosted.org/packages/fa/37/45185cb5abb30d7257104c434fe0b07e5a195a6847506c074527aa599ec/Click-7.0-py2.py3-none-any.whl
Collecting Werkzeug>=0.14 (from Flask==1.0.2->-r requirements.txt (line 1))
  Using cached https://files.pythonhosted.org/packages/ce/42/3aeda98f96e85fd26180534d36570e4d18108d62ae36f87694b476b83d6f/Werkzeug-0.16.0-py2.py3-none-any.whl
Collecting MarkupSafe>=0.23 (from Jinja2>=2.10->Flask==1.0.2->-r requirements.txt (line 1))
  Using cached https://files.pythonhosted.org/packages/98/7b/ff284bd8c80654e471b769062a9b43cc5d03e7a615048d96f4619df8d420/MarkupSafe-1.1.1-cp37m-cp37m-manylinux1_x86_64.whl
Installing collected packages: itsdangerous, MarkupSafe, Jinja2, click, Werkzeug, Flask
Successfully installed Flask-1.0.2 Jinja2-2.10.1 MarkupSafe-1.1.1 Werkzeug-0.16.0 click-7.0 itsdangerous-1.1.0
(venv) @cloudshell:~/python-appengine-test/RandomNumberGen-Servlets/RandomNumberPythonAE (python-ae-test-2) $
```

8. Run in the Flask development server with "python main.py"

```
(venv) @cloudshell:~/python-appengine-test/RandomNumberGen-Servlets/RandomNumberPythonAE (python-ae-test-2) $ python main.py
* Serving Flask app "main" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:8080/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 280-397-816
```

Follow the link to view the app on the Flask development server

Press CTRL+C to stop the app.

9. Create the app with "gcloud app create". Choose a region.

```
(venv) [redacted]@cloudshell:~/python-appengine-test/RandomNumberGen-Servlets/RandomNumberPythonAE (python-ae-test-2)$ gcloud app
create
You are creating an app for project [python-ae-test-2].
WARNING: Creating an App Engine application for a project is irreversible and the region
cannot be changed. More information about regions is at
<https://cloud.google.com/appengine/docs/locations>.

Please choose the region where you want your App Engine application
located:

[1] asia-east2      (supports standard and flexible)
[2] asia-northeast1 (supports standard and flexible)
[3] asia-northeast2 (supports standard and flexible)
[4] asia-south1     (supports standard and flexible)
[5] australia-southeast1 (supports standard and flexible)
[6] europe-west     (supports standard and flexible)
[7] europe-west2    (supports standard and flexible)
[8] europe-west3    (supports standard and flexible)
[9] europe-west6    (supports standard and flexible)
[10] northamerica-northeast1 (supports standard and flexible)
[11] southamerica-east1 (supports standard and flexible)
[12] us-central     (supports standard and flexible)
[13] us-east1       (supports standard and flexible)
[14] us-east4       (supports standard and flexible)
[15] us-west2       (supports standard and flexible)
[16] cancel

Please enter your numeric choice: 8

Creating App Engine application in project [python-ae-test-2] and region [europe-west3]...done.
Success! The app is now created. Please use 'gcloud app deploy' to deploy your first app.
```

10. Deploy the app with "gcloud app deploy app.yaml --project [Project ID]"

```
(venv) [redacted]@cloudshell:~/python-appengine-test/RandomNumberGen-Servlets/RandomNumberPythonAE (python-ae-test-2)$ gcloud app
deploy app.yaml --project python-ae-test-2
Services to deploy:

descriptor: [/home/[redacted]/python-appengine-test/RandomNumberGen-Servlets/RandomNumberPythonAE/app.yaml]
source:     [/home/[redacted]/python-appengine-test/RandomNumberGen-Servlets/RandomNumberPythonAE]
target project: [python-ae-test-2]
target service: [default]
target version: [20190930t173750]
target url:   [https://python-ae-test-2.appspot.com]

Do you want to continue (Y/n)? y

Beginning deployment of service [default]...

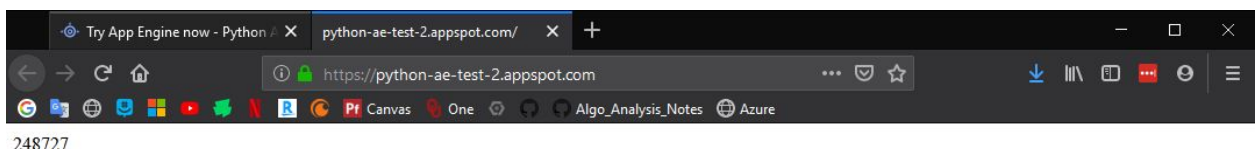
[Progress bar: Uploading 989 files to Google Cloud Storage]

File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://python-ae-test-2.appspot.com]

You can stream logs from the command line by running:
$ gcloud app logs tail -s default

To view your application in the web browser run:
$ gcloud app browse
```

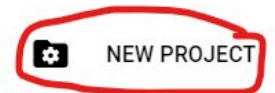
11. The URL of your app is shown in the output of step 10 under target url. Or you can use the "gcloud app browse" command to find the URL.



Python VM

1. Create a new project in Google Cloud Platform

Select a project



2. Navigate to the Compute Engine page and create a new VM instance with f1-micro, Ubuntu 18.04 LTS, and enable HTTP and HTTPS traffic

Name [?]

python-vm-experiment-2

Region [?] Zone [?]

europa-west3 (Frankfurt) europa-west3-a

Machine configuration [?]

Machine family

General-purpose Memory-optimized

Machine types for common workloads, optimized for cost and flexibility


Generation

First

Powered by Skylake CPU platform or one of its predecessors

Machine type

f1-micro (1 vCPU, 614 MB memory)

 vCPU Memory


1 shared core 614 MB

[CPU platform and GPU](#)

Container [?]

☐ Deploy a container image to this VM instance. [Learn more](#)

Boot disk [?]

 New 10 GB standard persistent disk

Image

Ubuntu 18.04 LTS

[Change](#)

Identity and API access [?]

Service account [?]

Compute Engine default service account

Access scopes [?]

☒ Allow default access

☐ Allow full access to all Cloud APIs


☐ Set access for each API

Firewall [?]

Add tags and firewall rules to allow specific network traffic from the Internet

☒ Allow HTTP traffic

☒ Allow HTTPS traffic

 You can use Cloud DNS to serve traffic for your domain from this VM.

[Show me how](#)

[Management, security, disks, networking, sole tenancy](#)

You will be billed for this instance. [Compute Engine pricing](#) [?]

[Create](#) [Cancel](#)

3. Connect to the instance via SSH



4. Execute the command "sudo apt update && sudo apt upgrade"

```
@python-vm-experiment-2:~$ sudo apt update && sudo apt upgrade
```

5. Type "hostname" to find hostname

```
@python-vm-experiment-2:~$ hostname  
python-vm-experiment-2
```

Edit the host file with "sudo nano /etc/hosts"

Under the localhost line, type the IP address of the server, press tab, then type the hostname

```
127.0.0.1 localhost  
34.89.244.250 python-vm-experiment-2  
  
# The following lines are desirable for IPv6 capable hosts  
::1 ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters  
ff02::3 ip6-allhosts  
169.254.169.254 metadata.google.internal metadata
```

6. Set up a firewall by executing the following commands:

- sudo apt install ufw
- sudo ufw default allow outgoing
- sudo ufw default deny incoming
- sudo ufw allow ssh
- sudo ufw allow http/tcp
- sudo ufw enable

```

@python-vm-experiment-2:~$ sudo apt install ufw
Reading package lists... Done
Building dependency tree
Reading state information... Done
ufw is already the newest version (0.36-0ubuntu0.18.04.1).
ufw set to manually installed.
The following packages were automatically installed and are no longer required:
  grub-pc-bin libnuma1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
@python-vm-experiment-2:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
@python-vm-experiment-2:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
@python-vm-experiment-2:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
@python-vm-experiment-2:~$ sudo ufw allow http/tcp
Rules updated
Rules updated (v6)
@python-vm-experiment-2:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
@python-vm-experiment-2:~$

```

7. Clone this repository with the command "git clone <https://github.com/Andy-Vu-Viz/RandomNumberGen-Servlets/>"

```

@python-vm-experiment-2:~$ git clone https://github.com/Andy-Vu-Viz/RandomNumberGen-Servlets/
Cloning into 'RandomNumberGen-Servlets'...
remote: Enumerating objects: 83, done.
remote: Counting objects: 100% (83/83), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 1056 (delta 23), reused 5 (delta 1), pack-reused 973
Receiving objects: 100% (1056/1056), 6.76 MiB | 9.33 MiB/s, done.
Resolving deltas: 100% (137/137), done.

```

8. Switch to the RandomNumberFlask directory with "cd RandomNumberGen-Servlets/RandomNumberFlask/"

```

@python-vm-experiment-2:~$ cd RandomNumberGen-Servlets/RandomNumberFlask/
@python-vm-experiment-2:~/RandomNumberGen-Servlets/RandomNumberFlask$

```

9. Install pip with "sudo apt install python3-pip"

```

@python-vm-experiment-2:~/RandomNumberGen-Servlets/RandomNumberFlask$ sudo apt install python3-pip

```

10. Install virtual environment with "sudo apt install python3-venv"

```

@python-vm-experiment-2:~/RandomNumberGen-Servlets/RandomNumberFlask$ sudo apt install python3-venv
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3.6-venv
The following NEW packages will be installed:
  python3-venv python3.6-venv
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 7392 B of archives.
After this operation, 44.0 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://europe-west3-a.gce.clouds.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 python3.6-venv amd64 3.6.8-1~18.04.2 [6184 B]
Get:2 http://europe-west3-a.gce.clouds.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 python3-venv amd64 3.6.7-1~18.04 [1208 B]
Fetched 7392 B in 0s (19.9 kB/s)
Selecting previously unselected package python3.6-venv.
(Reading database ... 66379 files and directories currently installed.)
Preparing to unpack .../python3.6-venv 3.6.8-1~18.04.2_amd64.deb ...
Unpacking python3.6-venv (3.6.8-1~18.04.2) ...
Selecting previously unselected package python3-venv.
Preparing to unpack .../python3-venv 3.6.7-1~18.04_amd64.deb ...
Unpacking python3-venv (3.6.7-1~18.04) ...
Setting up python3.6-venv (3.6.8-1~18.04.2) ...
Setting up python3-venv (3.6.7-1~18.04) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...

```

11. Create virtual environment with "python3 -m venv venv"

```

@python-vm-experiment-2:~/RandomNumberGen-Servlets/RandomNumberFlask$ python3 -m venv venv

```

12. Activate the virtual environment with "source venv/bin/activate"

```

@python-vm-experiment-2:~/RandomNumberGen-Servlets/RandomNumberFlask$ source venv/bin/activate
(venv) @python-vm-experiment-2:~/RandomNumberGen-Servlets/RandomNumberFlask$

```

13. Install project dependencies with "pip install -r requirements.txt"

```

(venv) @python-vm-experiment-2:~/RandomNumberGen-Servlets/RandomNumberFlask$ pip install -r requirements.txt
Collecting Click==7.0 (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/fa/37/45185cb5abbc30d7257104c434fe0b07e5a195a6847506c074527aa599ec/Click-7.0-py2.py3-none-any.whl (81kB)
    100% |#####| 81kB 8.1MB/s
Collecting Flask==1.1.1 (from -r requirements.txt (line 2))
  Downloading https://files.pythonhosted.org/packages/9b/93/628509b8d5dc749656a9641f4caf13540e2cdec85276964ff8f43bbbd3b/Flask-1.1.1-py2.py3-none-any.whl (94kB)
    100% |#####| 102kB 10.5MB/s
Collecting itsdangerous==1.1.0 (from -r requirements.txt (line 3))
  Downloading https://files.pythonhosted.org/packages/76/ae/44b03b253d6fade317f32c24d100b3b35c2239807046a4c953c7b99fa49e/itsdangerous-1.1.0-py2.py3-none-any.whl
Collecting Jinja2==2.10.1 (from -r requirements.txt (line 4))
  Downloading https://files.pythonhosted.org/packages/1d/e7/fd8b501e7a6dfe492a433deb7b9d833d39ca74916fa8bc63dd1a4947a671/Jinja2-2.10.1-py2.py3-none-any.whl (124kB)
    100% |#####| 133kB 8.7MB/s
Collecting MarkupSafe==1.1.1 (from -r requirements.txt (line 5))
  Downloading https://files.pythonhosted.org/packages/b2/5f/23e0023be6bb885d00ffbefad2942bc51a620328ee910f64abe5a8d18dd1/MarkupSafe-1.1.1-cp36-cp36m-manylinux1_x86_64.whl
Requirement already satisfied: pkg-resources==0.0.0 in ./venv/lib/python3.6/site-packages (from -r requirements.txt (line 6))
Collecting Werkzeug==0.15.6 (from -r requirements.txt (line 7))
  Downloading https://files.pythonhosted.org/packages/b7/61/c0a1ad9ad80db012ed7191af98fa05faa95fa09eceb71bb6fa8b66e6a43/Werkzeug-0.15.6-py2.py3-none-any.whl (328kB)
    100% |#####| 337kB 3.9MB/s
Installing collected packages: Click, itsdangerous, Werkzeug, MarkupSafe, Jinja2, Flask
Successfully installed Click-7.0 Flask-1.1.1 Jinja2-2.10.1 MarkupSafe-1.1.1 Werkzeug-0.15.6 itsdangerous-1.1.0

```

14. Install nginx with "sudo apt install nginx"

```

(venv) @python-vm-experiment-2:~/RandomNumberGen-Servlets/RandomNumberFlask$ sudo apt install nginx

```

15. Install gunicorn with "pip install gunicorn"

```
(venv) ~$ python-virtualenv-2:~/RandomNumberGen-Servlets/RandomNumberFlask$ pip install gunicorn
Collecting gunicorn
  Downloading https://files.pythonhosted.org/packages/8c/da/b8dd8deb741bfff556db53902d4706774c8e1e67265f69528c14c003644e6/gunicorn-19.9.0-py2.py3-none-any.whl (112kB)
    100% |#####| 122kB 9.1MB/s
Installing collected packages: gunicorn
Successfully installed gunicorn-19.9.0
```

16. Update nginx config file by doing the following:

```
sudo rm /etc/nginx/sites-enabled/default
```

```
sudo nano /etc/nginx/sites-enabled/random_number_generator
```

```
(venv) ~$ python-virtualenv-2:~/RandomNumberGen-Servlets/RandomNumberFlask$ sudo rm /etc/nginx/sites-enabled/default
(venv) ~$ python-virtualenv-2:~/RandomNumberGen-Servlets/RandomNumberFlask$ sudo nano /etc/nginx/sites-enabled/random_number_generator
```

Add this to /etc/nginx/sites-enabled/random_number_generator. Change server_name to your server's IP.

```
server {
    listen 80;
    server_name = 34.89.244.250;

    location / {
        proxy_pass http://localhost:8000;
        include /etc/nginx/proxy_params;
        proxy_redirect off;
    }
}
```

17. Restart nginx server with "sudo systemctl restart nginx"

```
(venv) ~$ python-virtualenv-2:~/RandomNumberGen-Servlets/RandomNumberFlask$ sudo systemctl restart nginx
```

18. Install supervisor with "sudo apt install supervisor"

```
(venv) ~$ python-virtualenv-2:~/RandomNumberGen-Servlets/RandomNumberFlask$ sudo apt install supervisor
```

19. Find out the number of workers for gunicorn with $(2 * \text{number of cores}) + 1$
(execute "nproc --all" to find number of cores)

```
(venv) ~$ python-virtualenv-2:~/RandomNumberGen-Servlets/RandomNumberFlask$ nproc --all
1
```

20. Set up supervisor with "sudo nano /etc/supervisor/conf.d/random_number_generator.conf"

```
(venv) python-vm-experiment-2:~/RandomNumberGen-Servlets/RandomNumberFlask$ sudo nano /etc/supervisor/conf.d/random_number_generator.conf
```

21. Add this to /etc/supervisor/conf.d/random_number_generator.conf. Change [Your Username] to your username and [Number of Workers] with the result of your number of workers calculation.

```
[program:random_number_generator]
directory=/home/[Your Username]/RandomNumberGen-Servlets/RandomNumberFlask
command=/home/[Your Username]/RandomNumberGen-Servlets/RandomNumberFlask/venv/bin/gunicorn -w [Number of Workers] random_number_generator:app
user=[Your Username]
autostart=true
autorestart=true
stopasgroup=true
killasgroup=true
```

22. Restart supervisor with "sudo supervisorctl reload"

```
(venv) python-vm-experiment-2:~/RandomNumberGen-Servlets/RandomNumberFlask$ sudo supervisorctl reload
Restarted supervisorctl
```

23. Setup a static IP for your virtual machine by going to the VPC Network page on GCP

Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
instance-1	us-central1-a			10.128.0.7 (nic0)	None	SSH
python-vm-	europa-west3-a	⚡ Increase perf.		10.156.0.2 (nic0)	35.242.194.148	SSH
	europa-			10.156.0.3	34.89.244.250	SSH

24. Under the "External IP addresses" find the IP address being used by the VM instance containing your deployment

Google Cloud Platform Python VM Test

VPC network External IP addresses

RESERVE STATIC ADDRESS REFRESH SHOW INFO PANEL

Filter resources

Name	External Address	Region	Type	Version	In use by	Network Tier	Labels
python-vm-experiment	35.242.194.148	europe-west3	Static	IPv4	VM instance python-vm-experiment (Zone europe-west3-a)	Premium	Change
test-4	34.70.25.80	us-central1	Static	IPv4	None	Premium	Change
-	34.89.244.250	europe-west3	Ephemeral	IPv4	VM instance python-vm-experiment-2 (Zone europe-west3-a)		

25. Switch "Ephemeral" to "Static" and reserve an IP with any name you find appropriate

<input type="checkbox"/>	python-vm-experiment-2	34.89.244.250	europe-west3	Static	IPv4	VM instance python-vm-experiment-2 (Zone europe-west3-a)	Premium	Change
--------------------------	------------------------	---------------	--------------	--------	------	--	---------	--------

26. To access the random number generator navigate to "http://[The IP Address of the Server]"

VPC network - Python VM Test 34.89.244.250/

34.89.244.250

996814

Our Deployments

Our original deployments can be found at the following URLs:

- Java App Engine <https://randomnumbergeneratorclean.appspot.com/>
- Java Virtual Machine <http://35.209.25.119/Servlets/RandomNumberGen>
- Python App Engine <http://python-rand-ae.appspot.com/>
- Python Virtual Machine <http://35.193.182.65/>

We were assigned the region europe-west3 zone a. These deployments can be found at

- Java App Engine <https://glassy-droplet-254017.appspot.com>
- Java Virtual Machine <http://35.198.133.185/Servlets/RandomNumberGen>
- Python App Engine <https://python-ae-test.appspot.com/>

- Python Virtual Machine <http://35.242.194.148/>

Testing

Our timing scripts can be found at <https://github.com/rsjk/RandomTiming/tree/master/Timing> and in our SharePoint Experiment folder. They require Python 3 to run. One script, `timing_experiment.py`, accepts a file of URLs/IPs separated by new lines as an argument. The URLs/IPs must be in the format:

- `[Region]_[zone]_[VM|app]_[Java|Python]@1.2.3.4`

URLs/IPs posted in the class discussion can be found in `class_urls.txt` in the repository. The script pings each server a 10 times and computes an average latency. Results are printed to the command line and written to a file called `"timing_experiment_test_log.txt"`. The results are of the format:

- `[from_ip_address]: [Region]_[zone]_[VM|app]_[Java|Python]@1.2.3.4 time
random_number`

To run the script, execute the following command `"python3 timing_experiment.py [name of file]"`.

Another script, `timing_experiment_csv.py`, stores the results in a csv rather than a txt file. The results are stored in a file called `"latency_log.csv"`. The csv has 3 columns: info, url, and latency(ms). Info contains the region, platform, and language information. URL contains the URLs/IPs. Latency(ms) contains the average latency from 10 pings in milliseconds. To run the script, execute the following command `"python3 timing_experiment_csv.py [name of file]"`.

If the scripts are unable to successfully get a number, the error message is printed instead of the time.

Timing Results

Results of running `timing_experiment.py` can be found in `timing_experiment_test_log.txt` in the repository and our SharePoint Experiment folder. We also ran `timing_experiment_csv.py` to get the timing results as a csv file. The log can be found in `latency_log.csv` in the repository and in our SharePoint folder. From the csv data, we made the following graphs of our results:

