**Objective**

**Set up Stackdriver monitoring and perform a simple load test using ApacheBench or a similar tool.**

UPDATE: "StackDriver" has been replaced, retaining all past functionality, with the GC operation suite.

\*Google Cloud operation Suite (<https://cloud.google.com/products/operations>).

\*Benchmarking Tool Info: (<https://www.datadoghq.com/blog/apachebench/>)

\*Setting up Monitoring/Logging Agents: (<https://www.cloudskillsboost.google/focuses/10599?parent=catalog>)

**Step 1: Select your Project**

From the GCP console either create a new project or select an existing one.

**Step 2: Create a Compute Engine VM Instance**

Navigate to the Compute Engine tab and click on “Create Instance''. Choose your machine type (CPU etc) and make sure for the firewall you allow HTTP/HTTPS traffic. Wait until you see a green check mark status to let you know the instance has launched.

**Step 3: Add Apache2 HTTP Server to the instance**

From the project console page click on the SSH icon to open a terminal to connect to the instance.

Run the following commands on the terminal:

| sudo apt-get update |
| --- |

| sudo apt-get install apache2 php7.0 |
| --- |

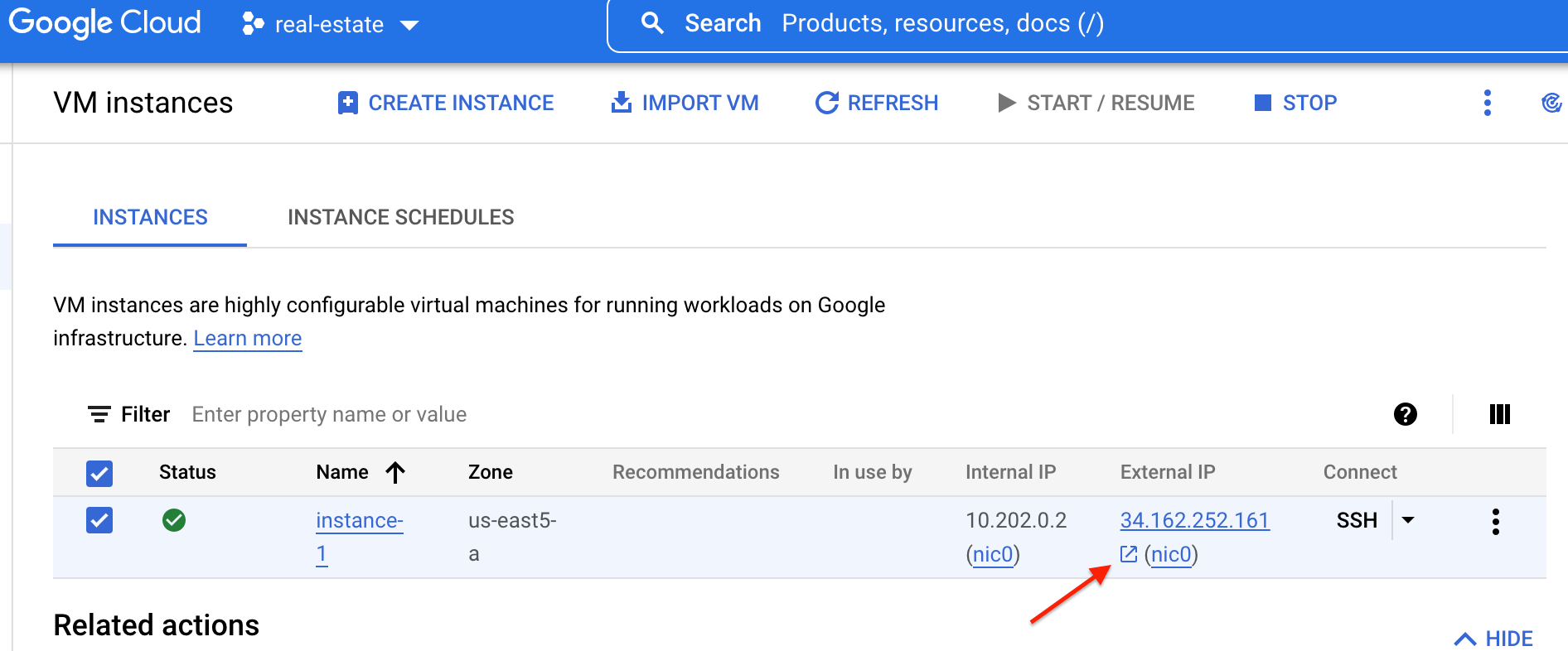
| sudo apt-get install -y apache2-utils |
| --- |

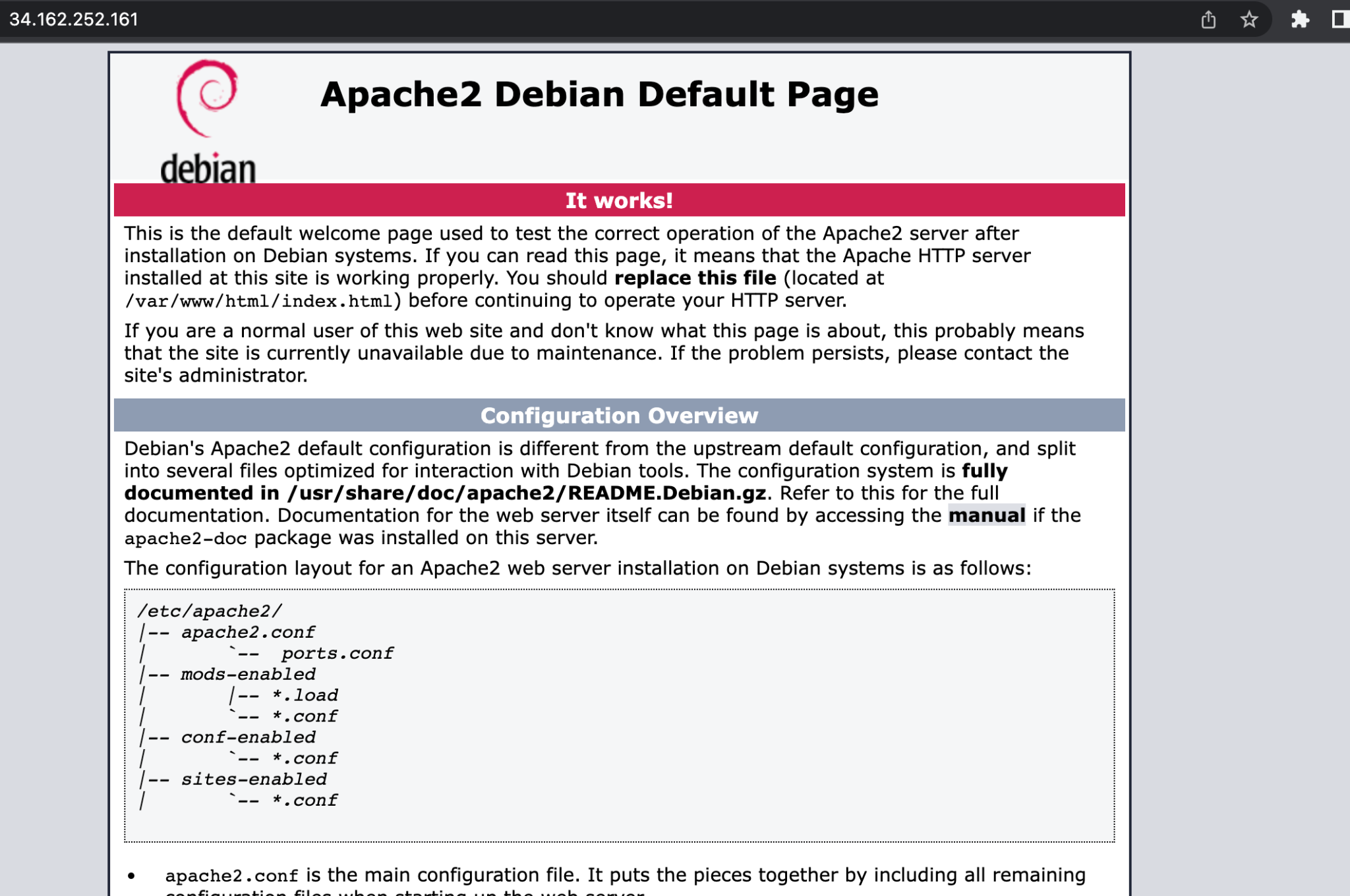
When asked if you want to continue, enter **Y**.

| sudo service apache2 restart |
| --- |

**Step 4: Check External IP for Apache2 Default Page**

Return to the Cloud Console, on the VM instances page. Click the External IP for lamp-1-vm instance to see the Apache2 default page for this instance.



Once clicking the External IP from the console page it should open up the Apache2 Default page letting you know it works.

**Step 5: Install the Monitoring Agents**

Set up a Monitoring Metric Scope by navigating to the Monitoring page in the console. Through the Overview page you will be able to set up alerts.

Run the Monitoring agent install script command in the SSH terminal of the VM instance to install the Cloud Monitoring agent through the following commands:

| curl -sSO https://dl.google.com/cloudagents/add-google-cloud-ops-agent-repo.sh |
| --- |

| sudo bash add-google-cloud-ops-agent-repo.sh --also-install |
| --- |

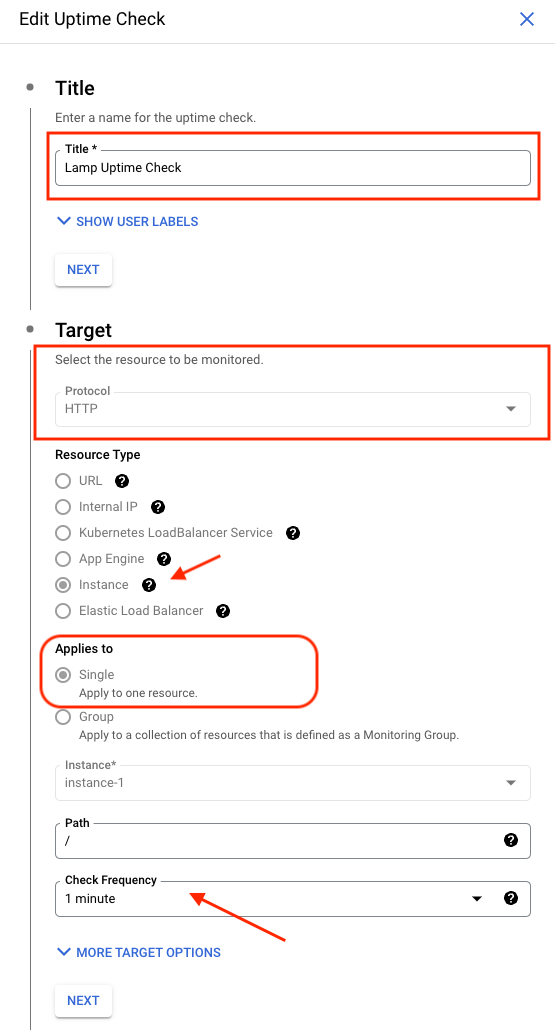
**Step 6: Install the Logging Agents**

Run the Logging agent install script command in the SSH terminal of the VM instance to install the Cloud Logging agent through the following commands:

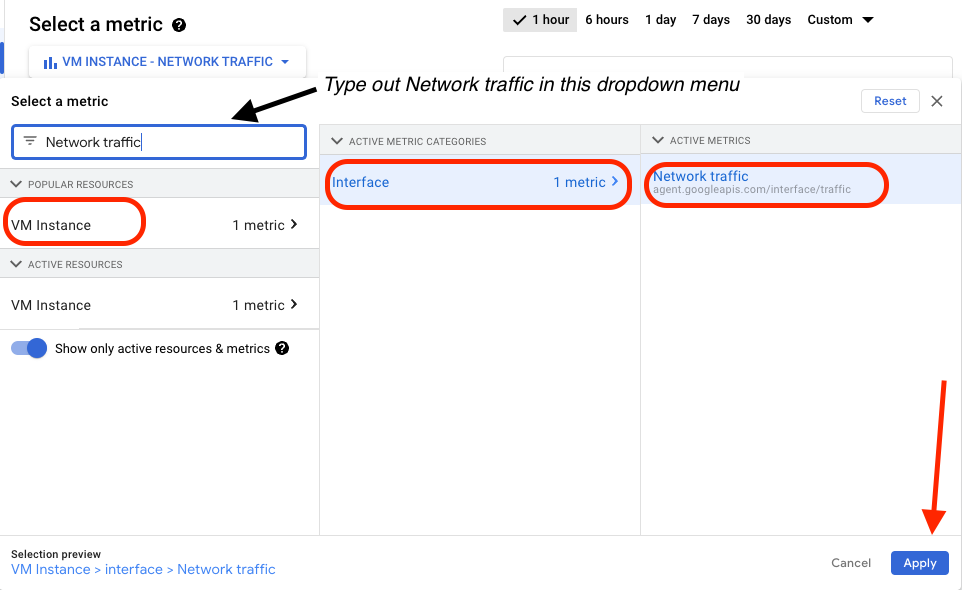
| sudo systemctl status google-cloud-ops-agent"\*" |
| --- |

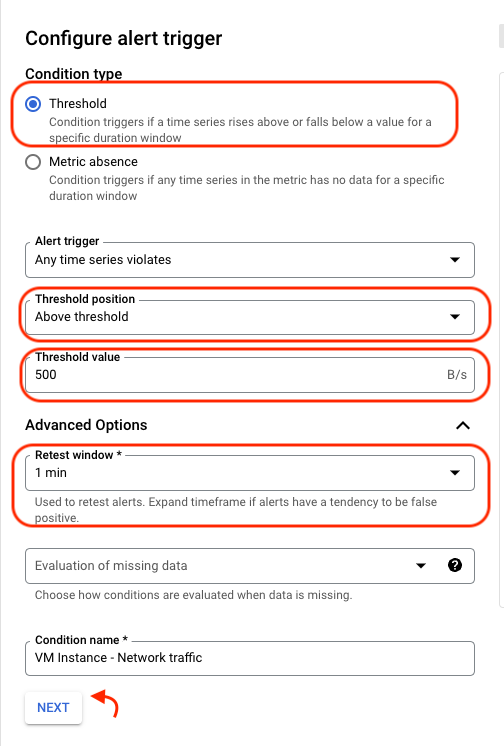
| sudo apt-get update |
| --- |

**Step 7: Create an uptime check**

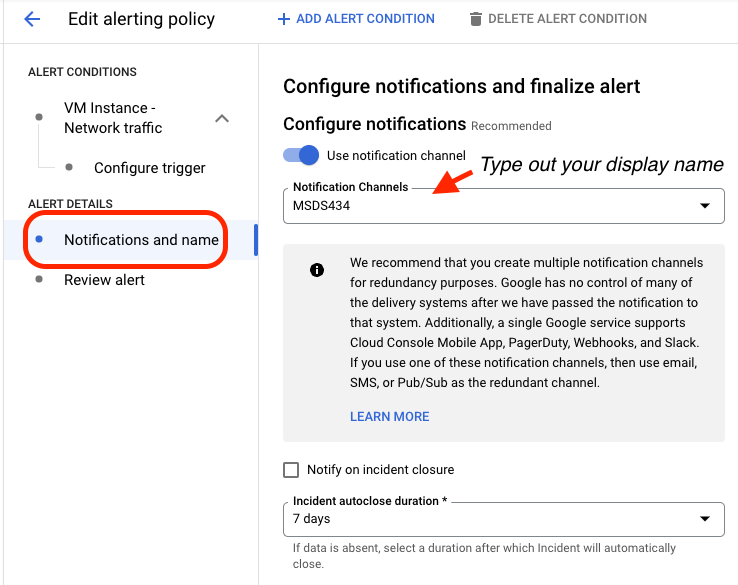
To confirm that the VM is always accessible, create an uptime check through the Monitoring page Menu and set the following parameters. Click on Test to verify that your uptime check can connect to the VM when you get a green check mark, click Create.

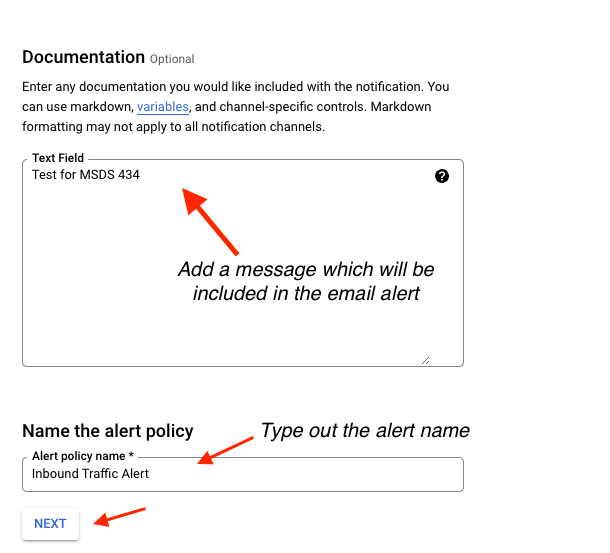
**Step 8: Create an Alerting Policy**

Create an alerting policy by navigating to the Monitoring page Menu and selecting the Alerting page. Through the Alerting page click on the CREATE POLICY icon and set up the following fields when selecting the metrics. 



After applying the field above, click next to set your condition type and configure the alert trigger. Apply the following fields (highlighted in red) →

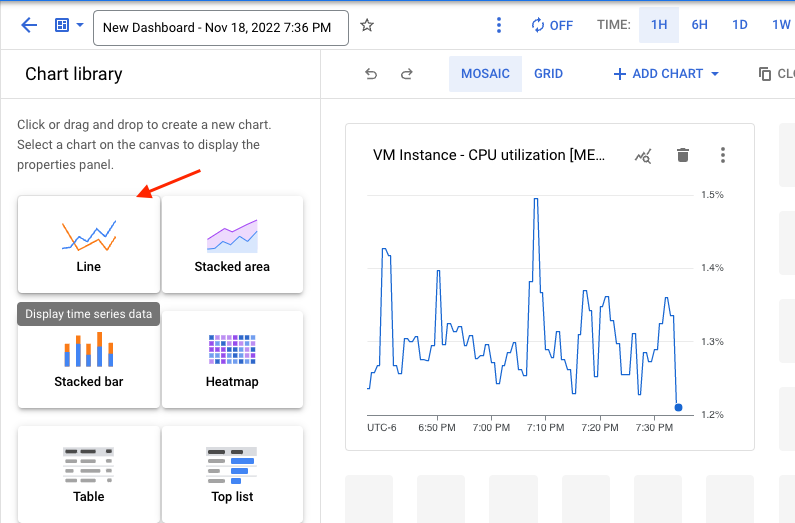


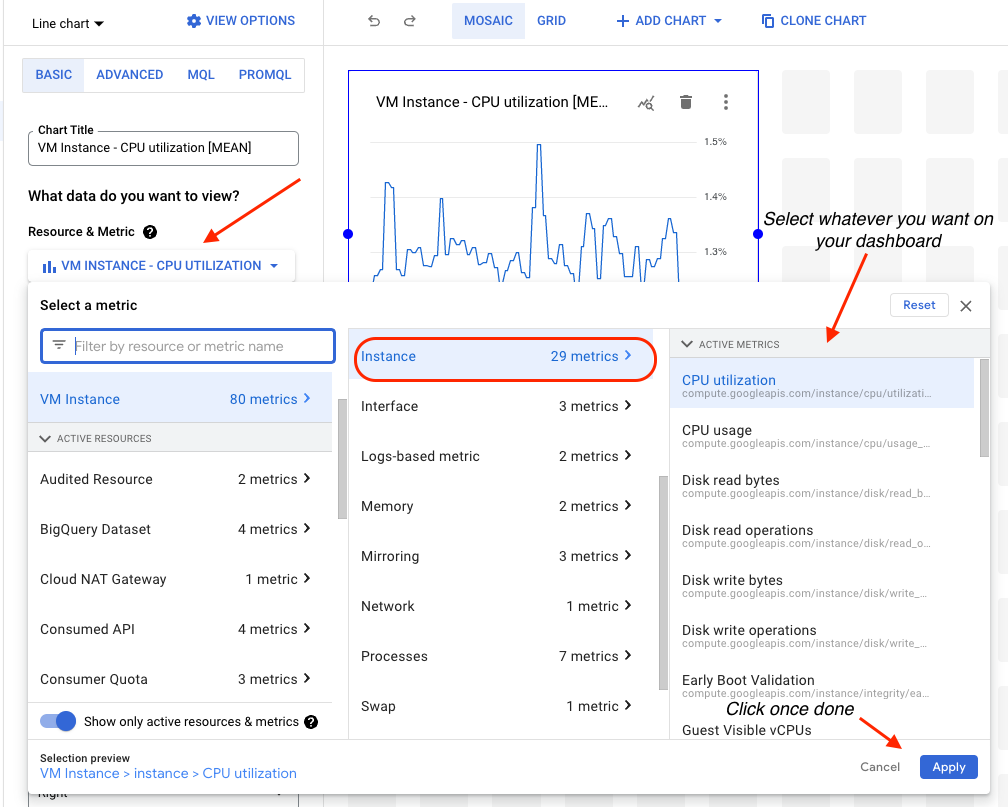


After reviewing the alert click on CREATE POLICY.

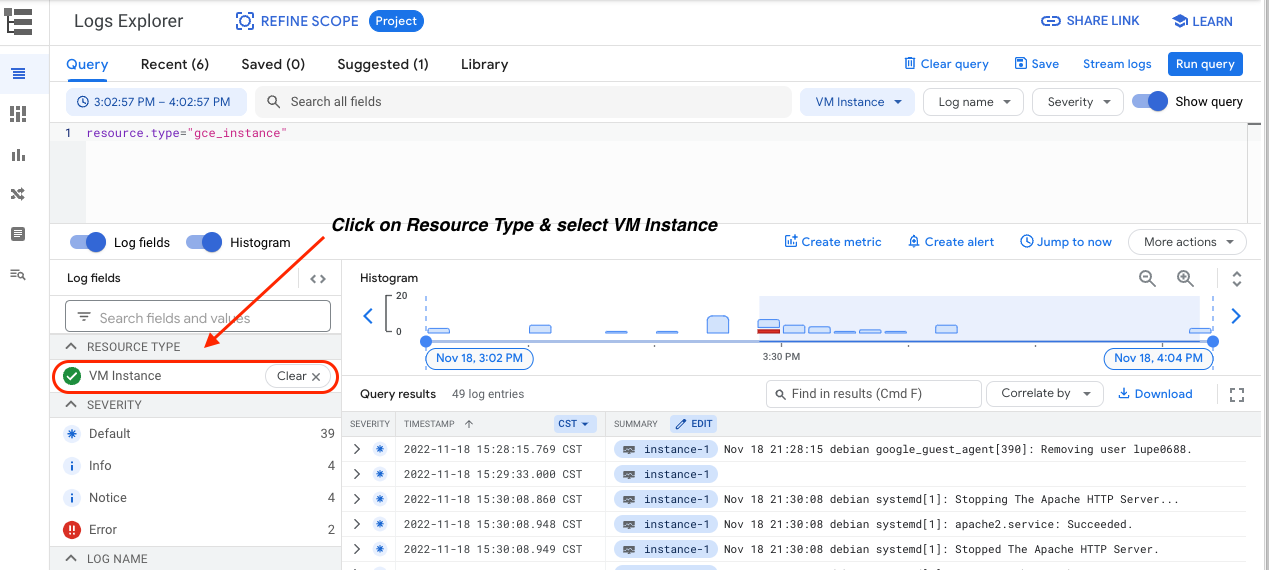
**Step 9: Create a Dashboard**

Create a dashboard to visualize your metrics by navigating to the Monitoring page Menu and selecting the Dashboards page. Through the Dashboard page click on the +CREATE DASHBOARD icon and complete the setup by following the example below.





**Step 10: Monitor your Logs**

To monitor the logs on the VM instance navigate to the Logging page from the Monitor navigation menu and select Logs Explorer.  ****

**Step 11: Update your VM**

With any virtual machine you will want to update it with the following command

| sudo apt update |
| --- |

**Step 12: Install Git**

| sudo apt install git |
| --- |

**Step 13: Clone your Github Repo to your instance**

| git clone https://github.com/GoogleCloudPlatform/python-docs-samples.git |
| --- |

**Step 14: Change into the directory where your app file is located**

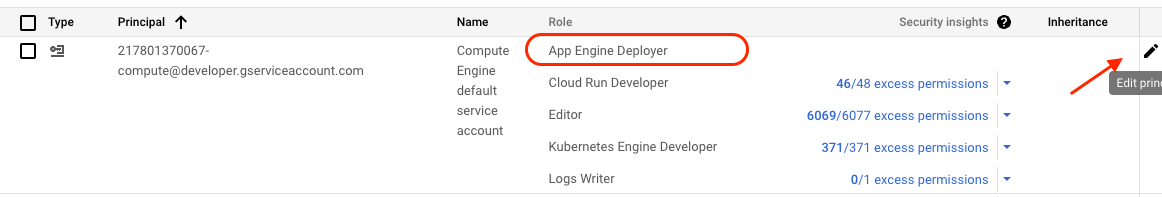
| cd python-docs-samples/ |
| --- |

| cd appengine/standard\_python3/hello\_world |
| --- |

| ls |
| --- |

**Step 15: Deploy your App**

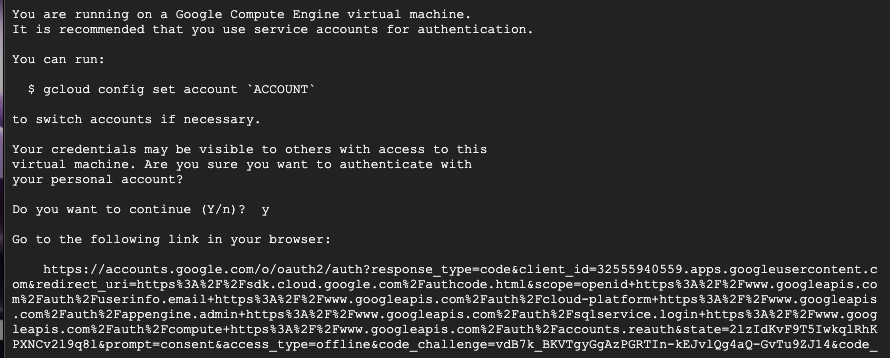
Prior to deploying your app ensure that you have permission to view applications on the project and that 217801370067-compute@developer.gserviceaccount.com has the App Engine Deployer (roles/appengine.deployer) role. If not then edit to add that role to that principle, see below:



| gcloud app deploy app.yaml |
| --- |

\*If you get an error again regarding having permission to view applications on the project. You can authorize with the following command:

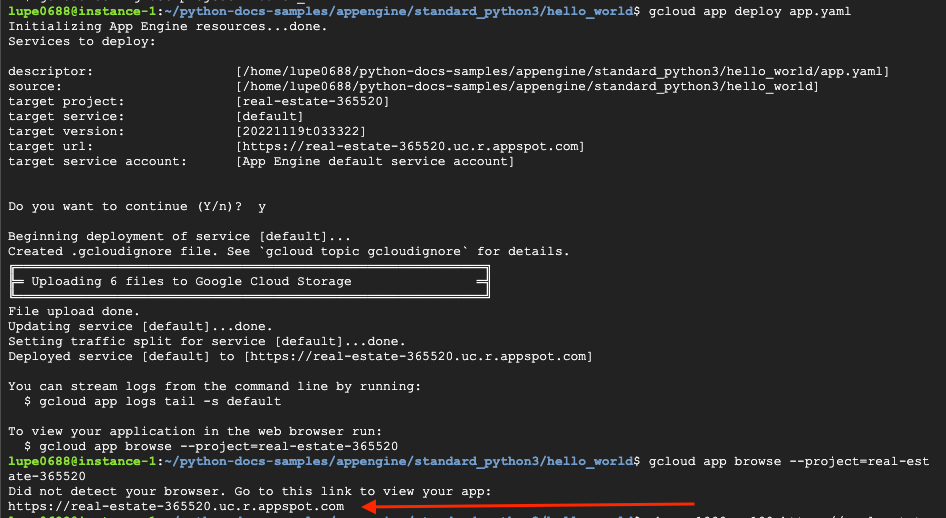
| gcloud auth login |
| --- |

****

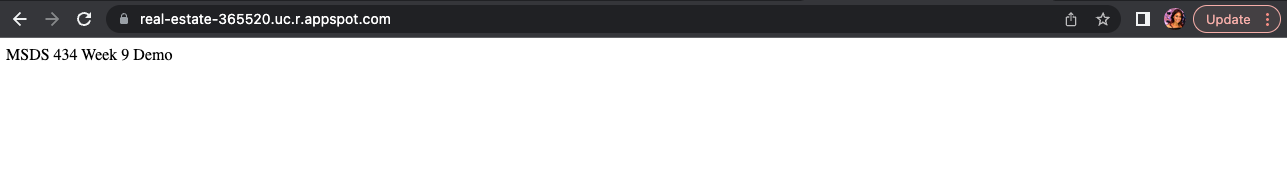
Copy the link to your browser and it will prompt you to sign in and redirect you to the SDK page to provide you with a verification code that you will enter into the terminal



| gcloud app deploy app.yaml |
| --- |



Result:



**Step 16: Perform Web Server Performance Testing**

Using ApacheBench ‘ab’ we can send a request argument

* -n: The number of requests to send
* -c: The number of concurrent requests to make

Template: ab -n 100000 -c 1000 <SERVER\_ADDRESS>

| ab -n 1000 -c100 https://real-estate-365520.uc.r.appspot.com/ |
| --- |

