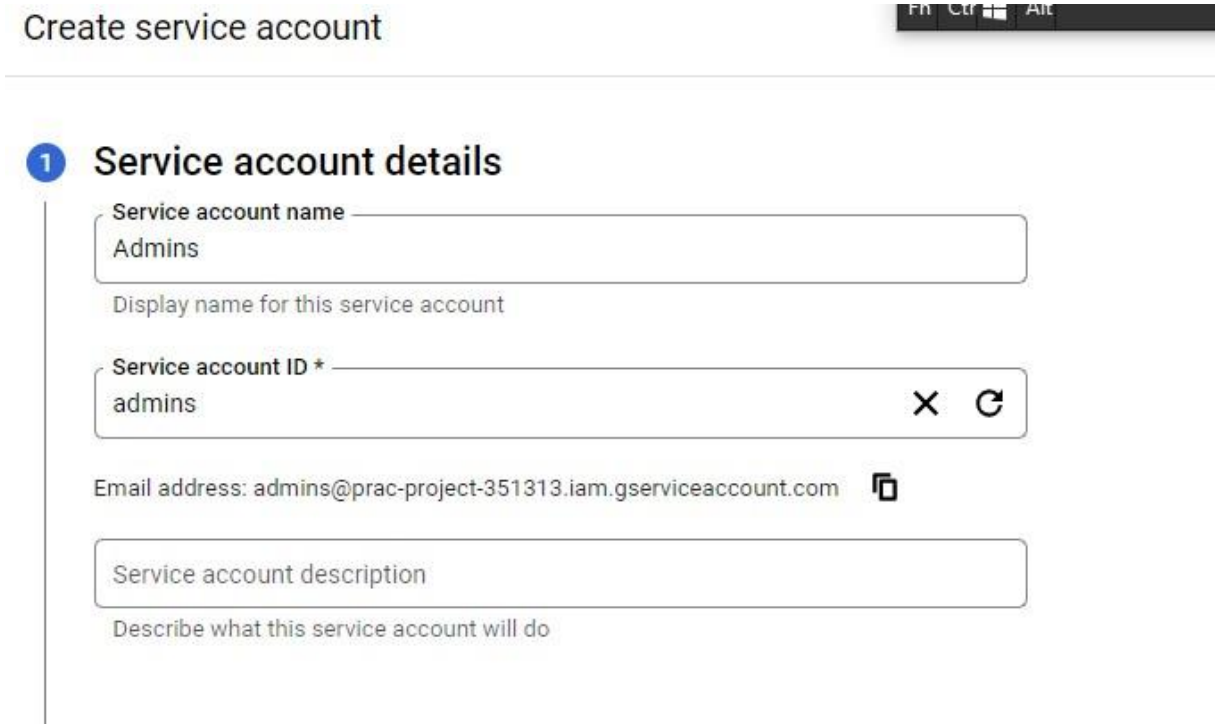
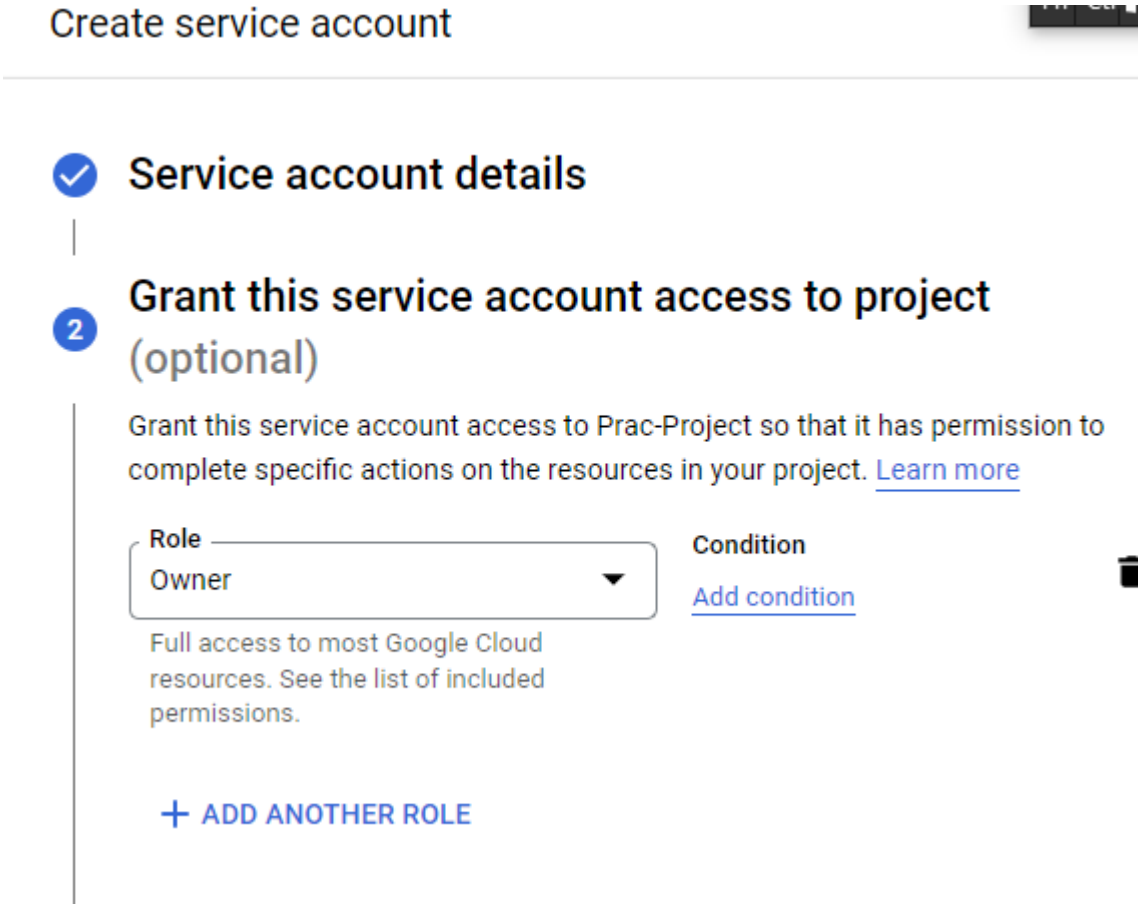
GCP Project

Create three users for Owner, Editor and Viewer role.

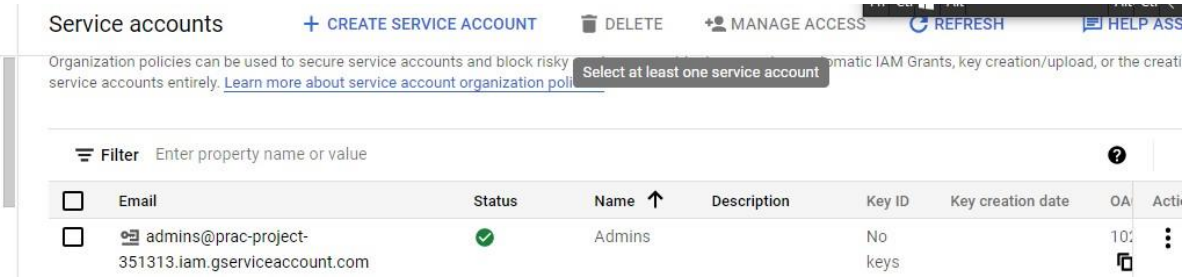
Go to service account.



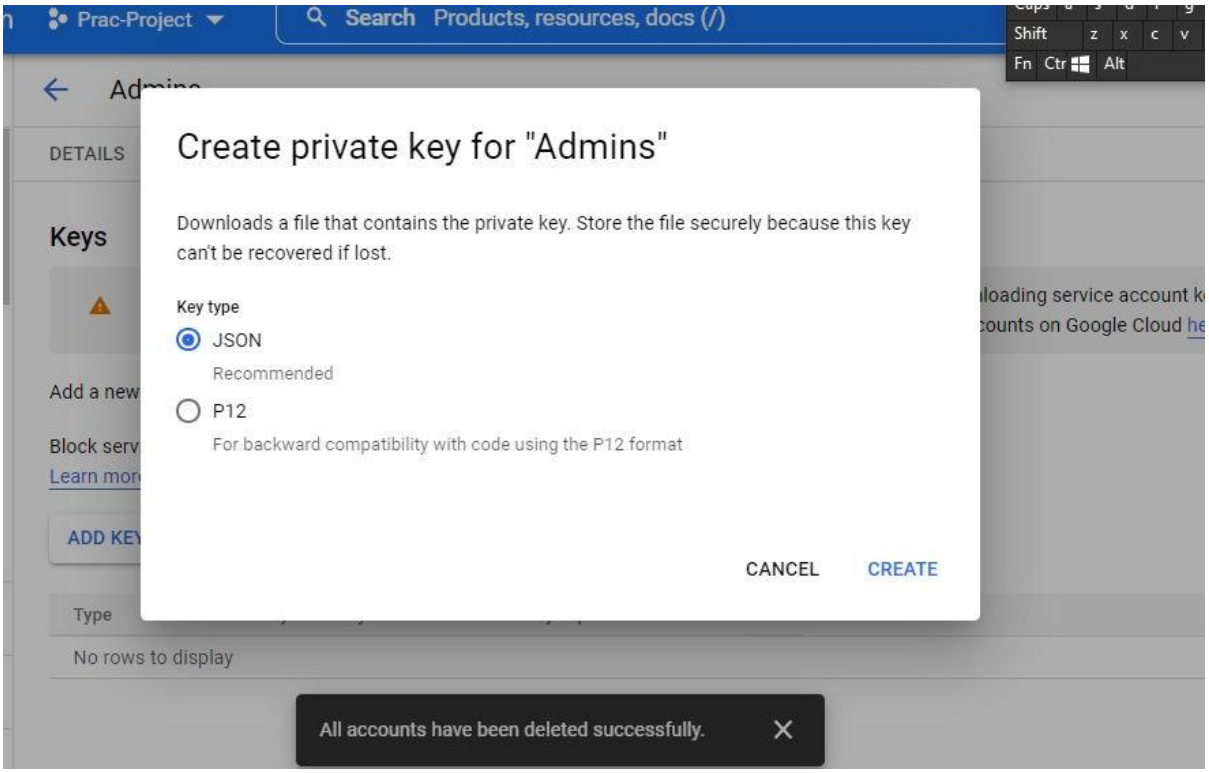
Select role.

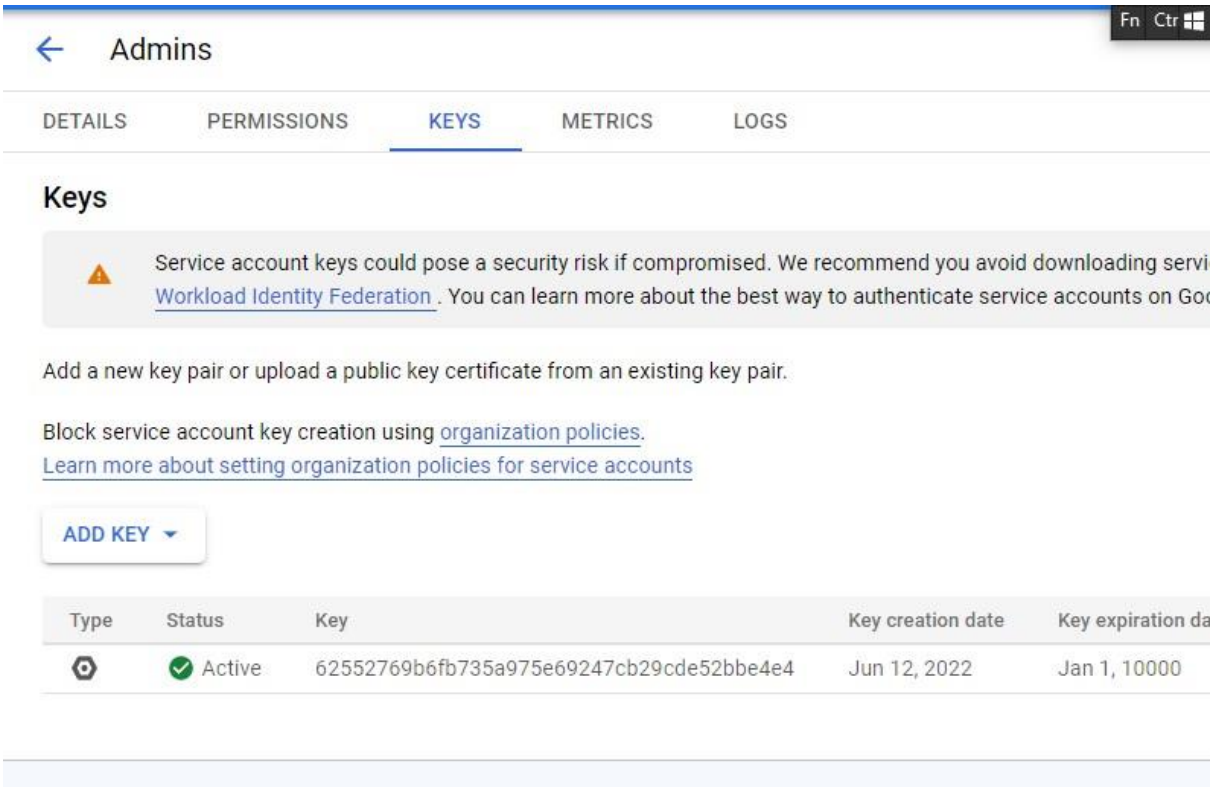


User created.

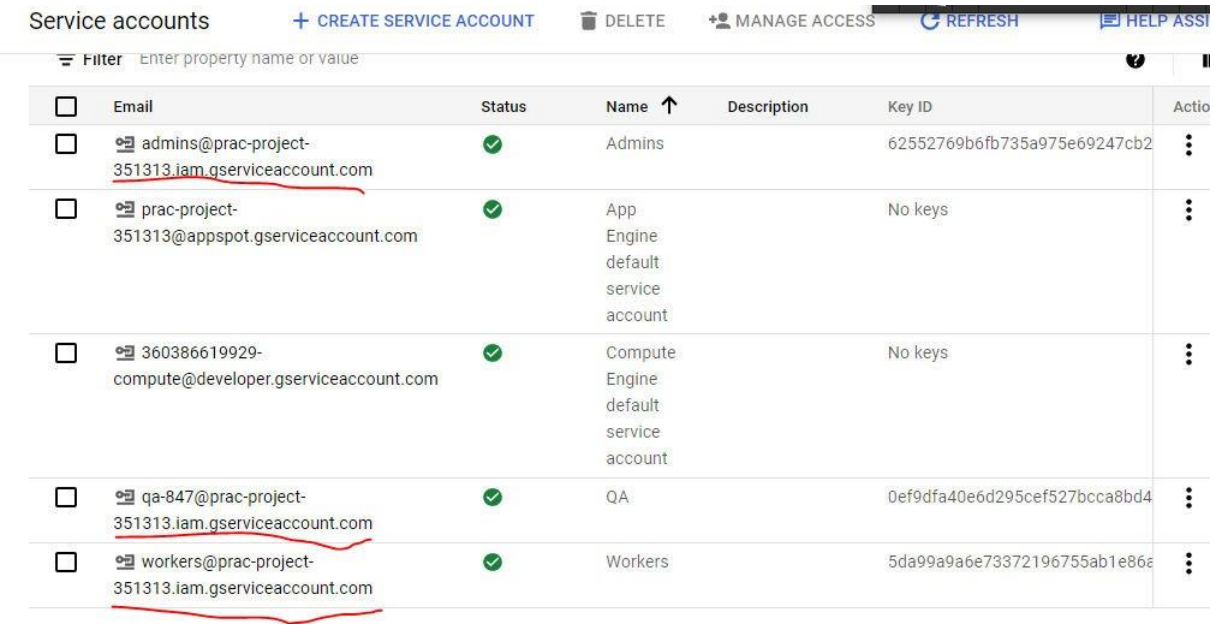


Generate key.

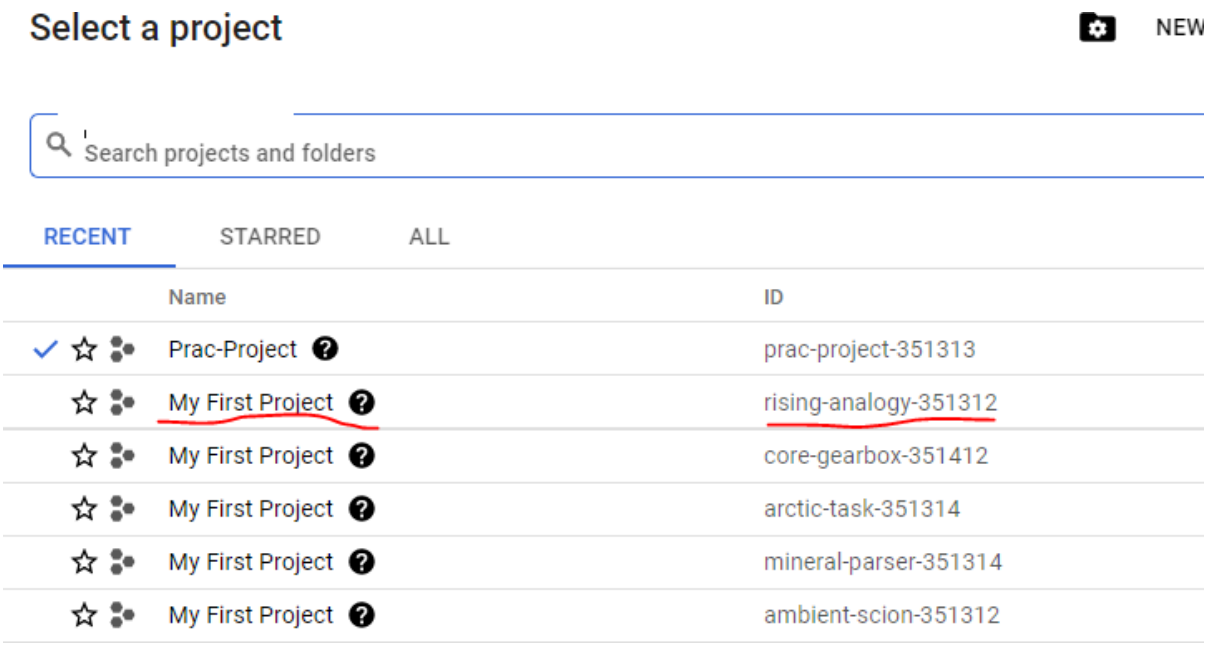




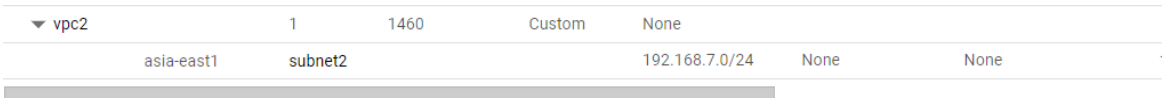
Similarly, Create QA and Workers.



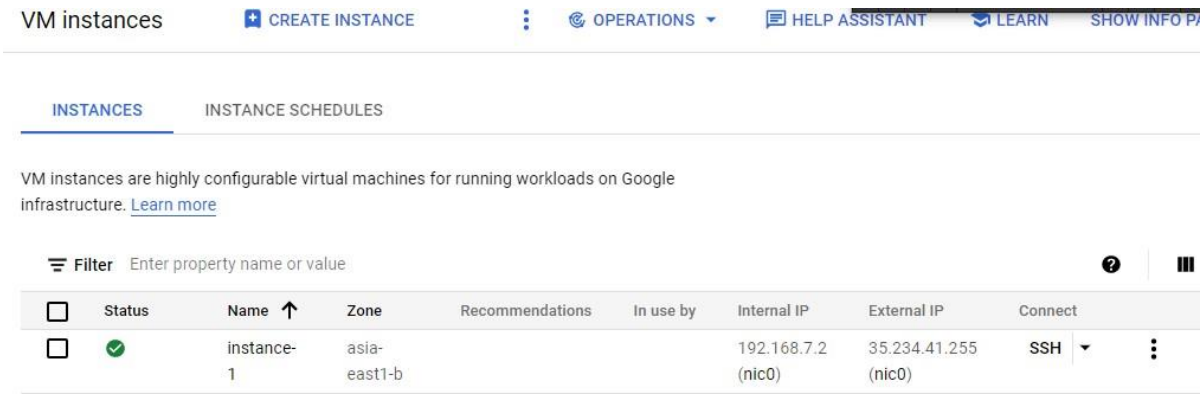
Now, We need to create one project and create a server .



Create a VPC.



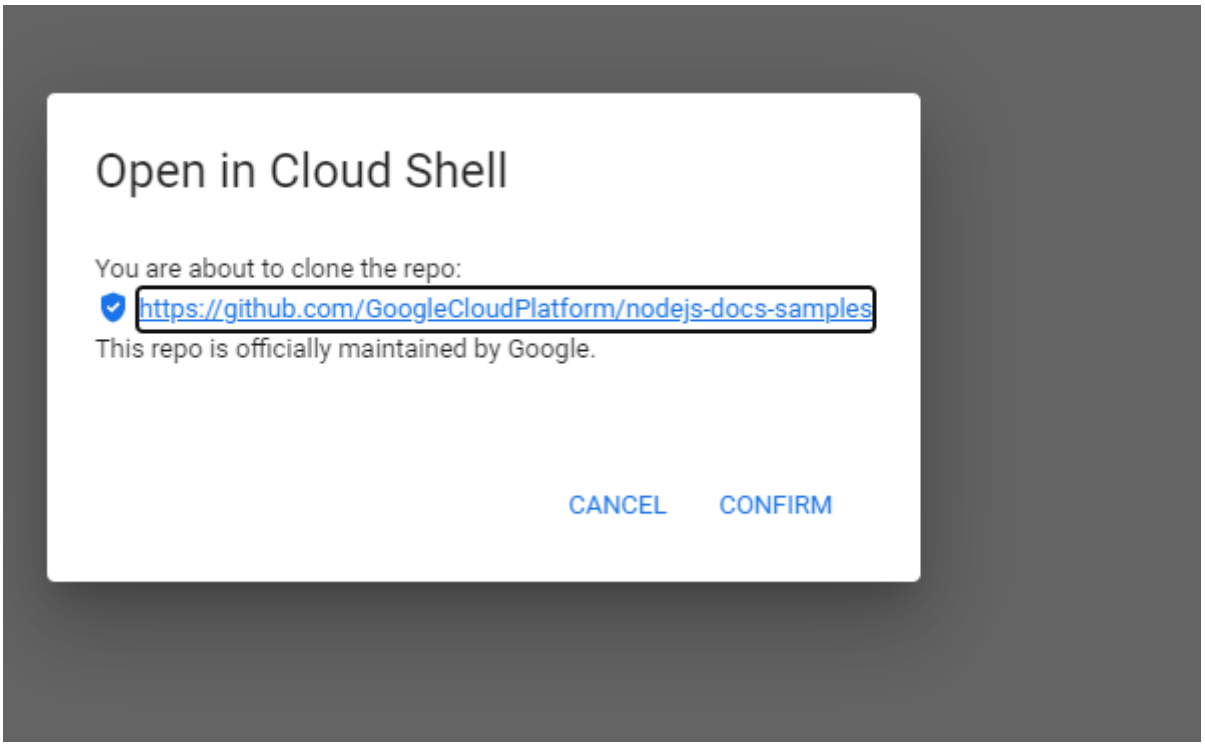
Create a VM inside that VPC.



1. Deploy the application, on app service which can be scaled up and down from 2 instances to 10 instances using apache tools.

Go to gcloud.

Clone the below repository.



Go inside the below folder.

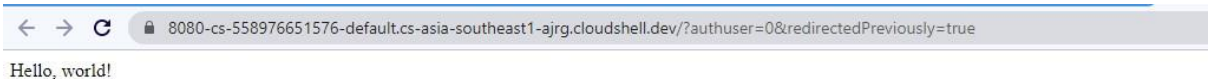
```
arpitacs0024@cloudshell:~/cloudshell_open/nodejs-docs-samples-7$ cd appengine/hello-world/standard
arpitacs0024@cloudshell:~/cloudshell_open/nodejs-docs-samples-7/appengine/hello-world/standard$ ls
app.js  app.yaml  package.json  README.md  test
arpitacs0024@cloudshell:~/cloudshell_open/nodejs-docs-samples-7/appengine/hello-world/standard$
```

Install dependencies.

```
arpitacs0024@cloudshell:~/cloudshell_open/nodejs-docs-samples-7/appengine/hello-world/standard$ npm install
(                    ) ⠧ idealTree:standard: sill idealTree buildDeps
```

```
found 0 vulnerabilities
arpitacs0024@cloudshell:~/cloudshell_open/nodejs-docs-samples-7/appengine/hello-world/standard$ npm start

> appengine-hello-world@0.0.2 start
> node app.js

App listening on port 8080
Press Ctrl+C to quit.
```

Web preview.

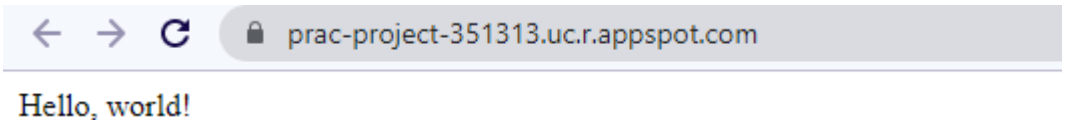


Deploy the Hello World app on App Engine by running the following command.

```
arpitacs0024@cloudshell:~/cloudshell_open/nodejs-docs-samples-7/appengine/hello-world/standard (prac-project-351313)$ gcloud app deploy
Services to deploy:

descriptor:                  [/home/arpitacs0024/cloudshell_open/nodejs-docs-samples-7/appengine/hello-world/standard/app.yaml]
source:                      [/home/arpitacs0024/cloudshell_open/nodejs-docs-samples-7/appengine/hello-world/standard]
target project:              [prac-project-351313]
target service:              [default]
target version:              [20220612t085654]
target url:                  [https://prac-project-351313.uc.r.appspot.com]
target service account:      [App Engine default service account]


Do you want to continue (Y/n)?

Beginning deployment of service [default]...
Created .gcloudignore file. See `gcloud topic gcloudignore` for details.
Uploading 1 file to Google Cloud Storage
100%
100%
File upload done.
Updating service [default]...working
```

Browse the application.

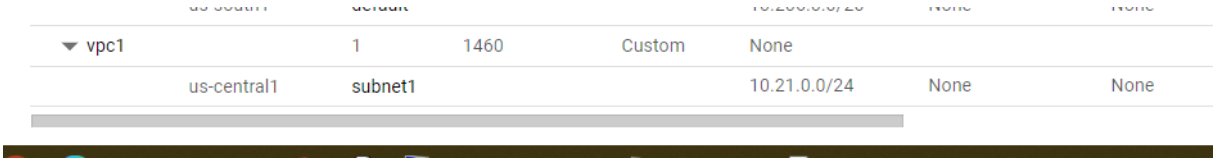
```
To view your application in the web browser run:
  $ gcloud app browse
arpitacs0024@cloudshell:~/cloudshell_open/nodejs-docs-samples-7/appengine/hello-world/standard (prac-project-351313)$ gcloud app browse
Did not detect your browser. Go to this link to view your app:
https://prac-project-351313.uc.r.appspot.com
arpitacs0024@cloudshell:~/cloudshell_open/nodejs-docs-samples-7/appengine/hello-world/standard (prac-project-351313)$
```



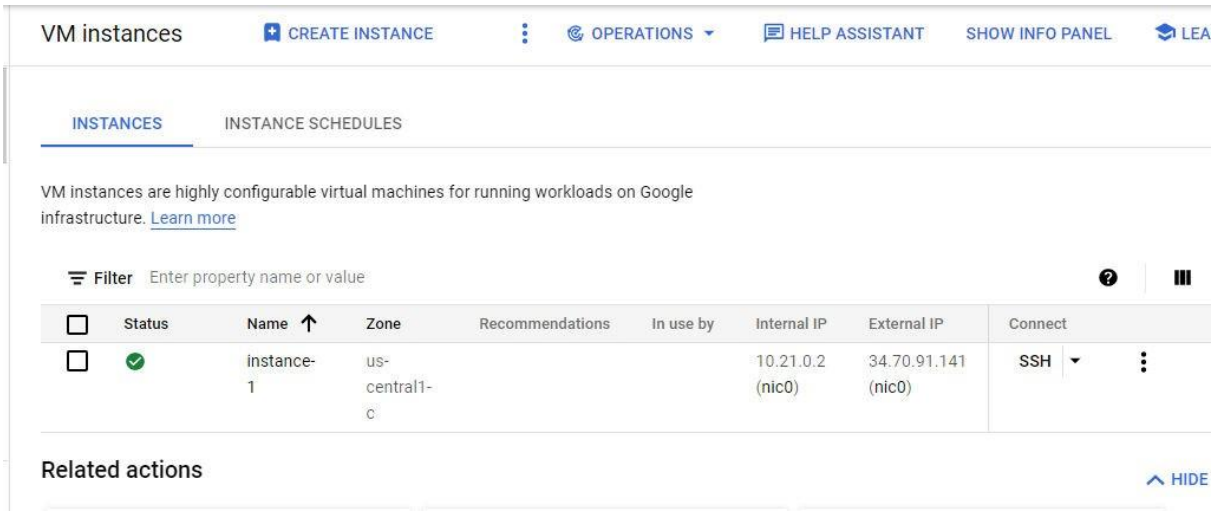
Hello, world!

We need to create instance in same region as app engine.

Create a VPC in first project.



Instance created.



Update the VM and install apache utils.

```
Reading package lists... Done
arpitacs0024@instance-1:~$ sudo apt-get install apache2-utils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libapr1 libaprutil1
The following NEW packages will be installed:
  apache2-utils libapr1 libaprutil1
0 upgraded, 3 newly installed, 0 to remove and 7 not upgraded.
Need to get 456 kB of archives.
After this operation, 1089 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

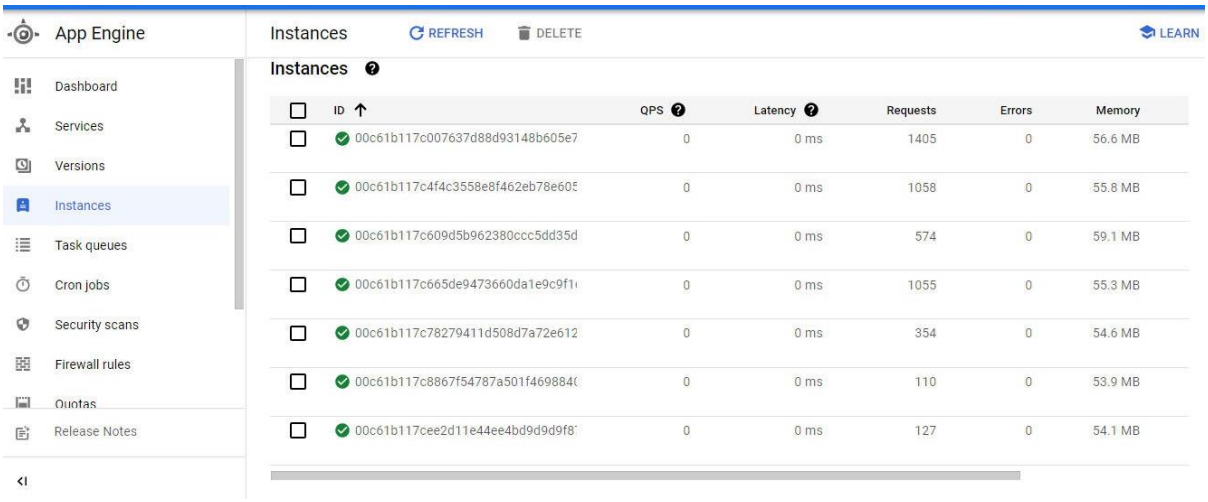
Run the below command for 5000 requests- Auto scaling

```
arpitacs0024@instance-1:~$ ab -n 5000 -c 1000 https://prac-project-351313.uc.r.appspot.com/
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking prac-project-351313.uc.r.appspot.com (be patient)
Completed 500 requests
Completed 1000 requests
Completed 1500 requests
Completed 2000 requests
Completed 2500 requests
Completed 3000 requests
Completed 3500 requests
Completed 4000 requests
Completed 4500 requests
```

Multiple requests .

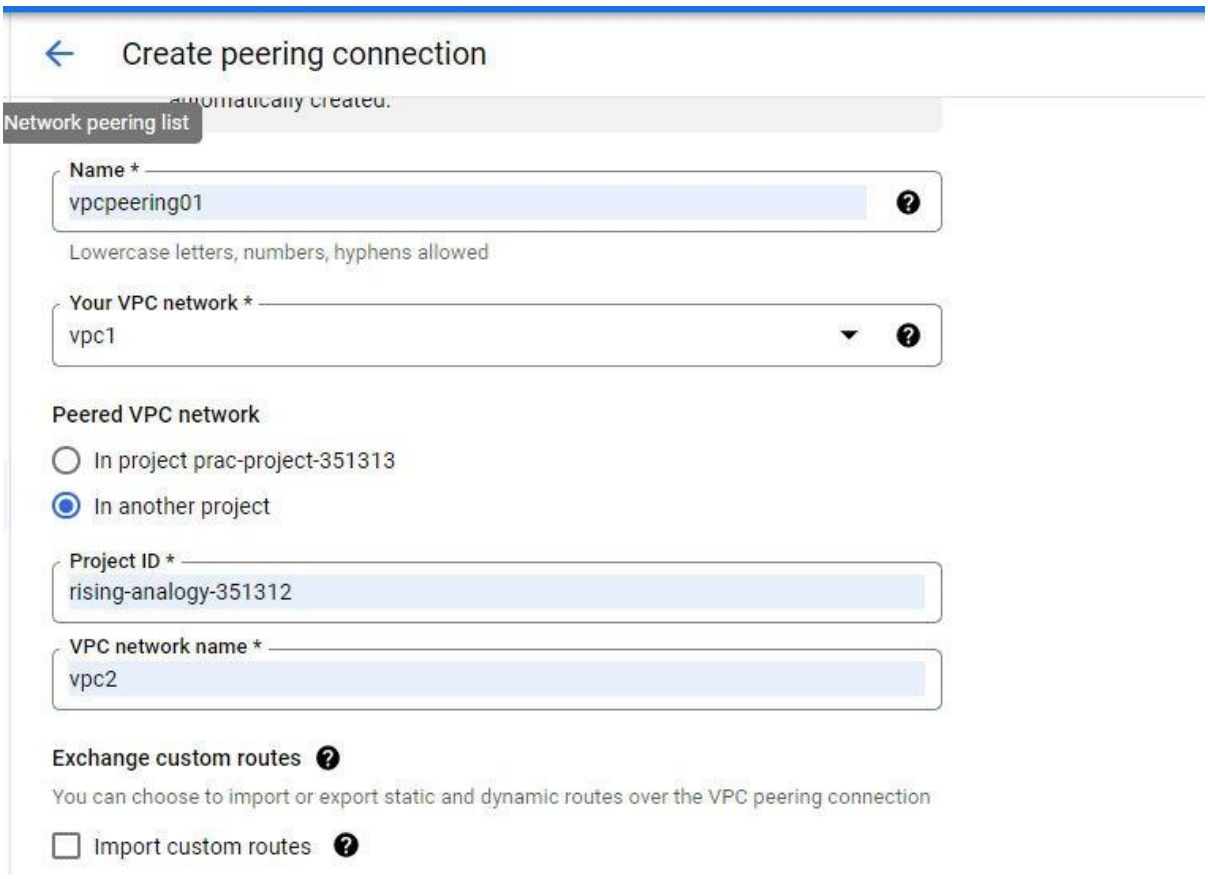
```
Every 2.0s: gcloud app instances list                                   cs-558976651576-default: Tue Jun  7 16:04:07 2022

SERVICE: default
VERSION: ver1
ID: 00c61b117ca63c99f55f0de492509277fe45517b6c7f20736532552633daeff1b03d28f3e2bfefeadb92ea46b4ae62cc507d4e11e4b7686a
VM_STATUS: N/A
VM_LIVENESS:
DEBUG_MODE:

SERVICE: default
VERSION: ver2
ID: 00c61b117c007637d88d93148b605e750301dc91b2c849df4b20f0bb553c7bbc9de9e1247cf41fd9ff843b8724ff2b48493fd9450067105f28e2cf
VM_STATUS: N/A
VM_LIVENESS:
DEBUG_MODE:

SERVICE: default
VERSION: ver2
ID: 00c61b117c4f4c3558e8f462eb78e605e9f9387027edd168959c2bf76a5abd1936013c5dd81f873a83db81de344dde79d9f1149cdb5f5d8dd4ea73
```

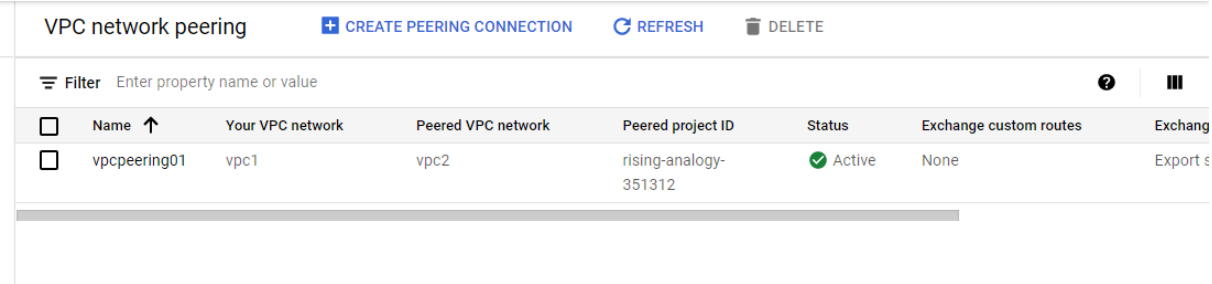
Multiple instances provisioned.



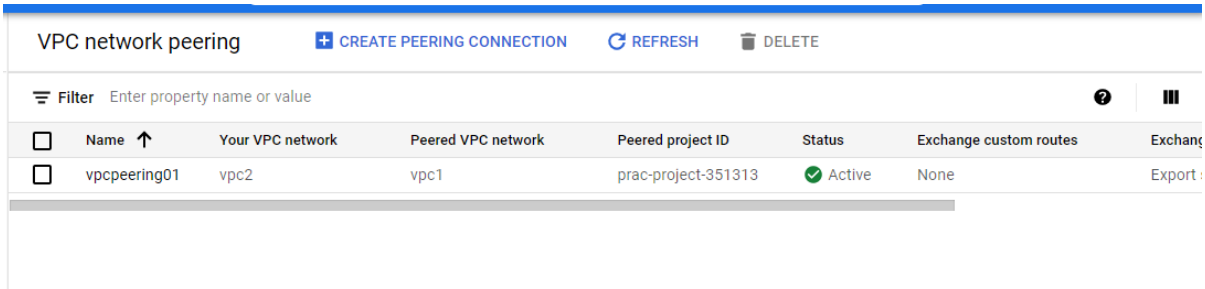
2. Now, We need to configure vpc peering so that VM of different project VPC able to connect.

In 1st project

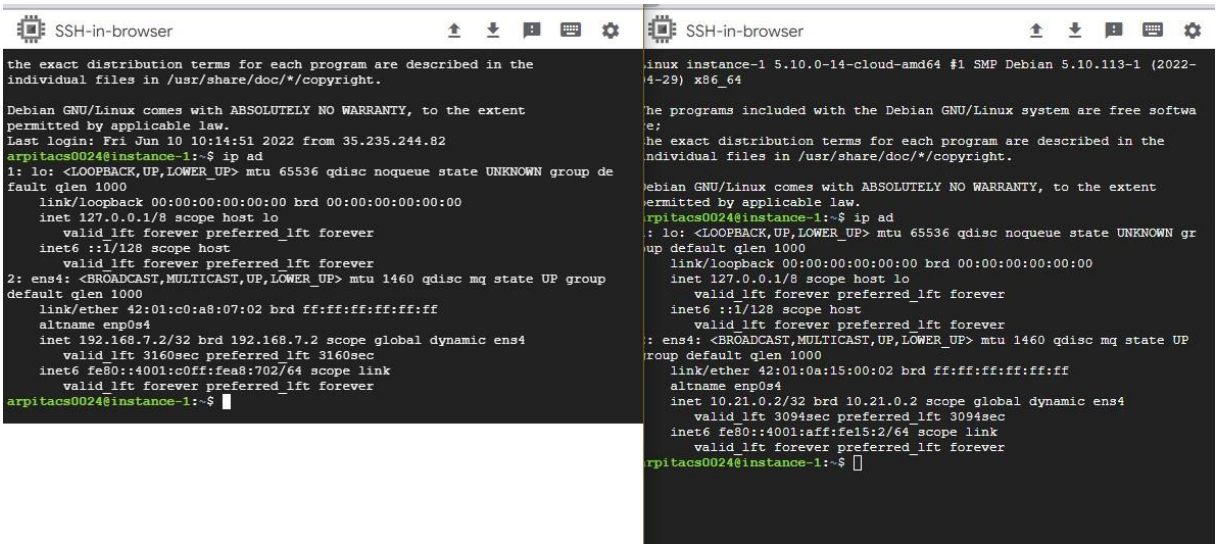




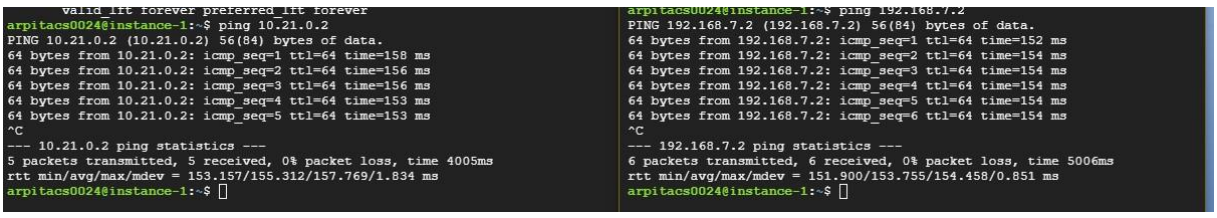
Similarly, Create in 2nd project.



Try to ping one vm from another vm.

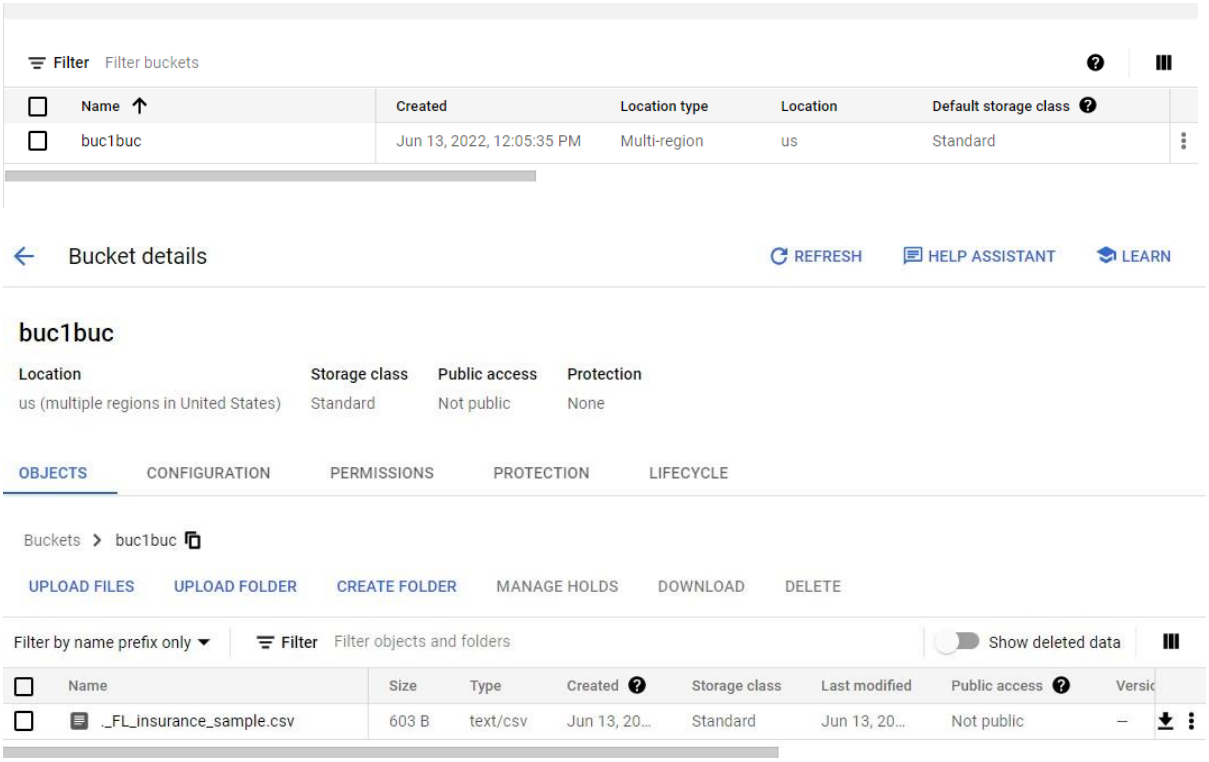


Able to ping both vm from each other.



3. The app service will be interacting with a database, for now the database exists in a local system, here is the exported database . Use this and import in Cloud SQL.

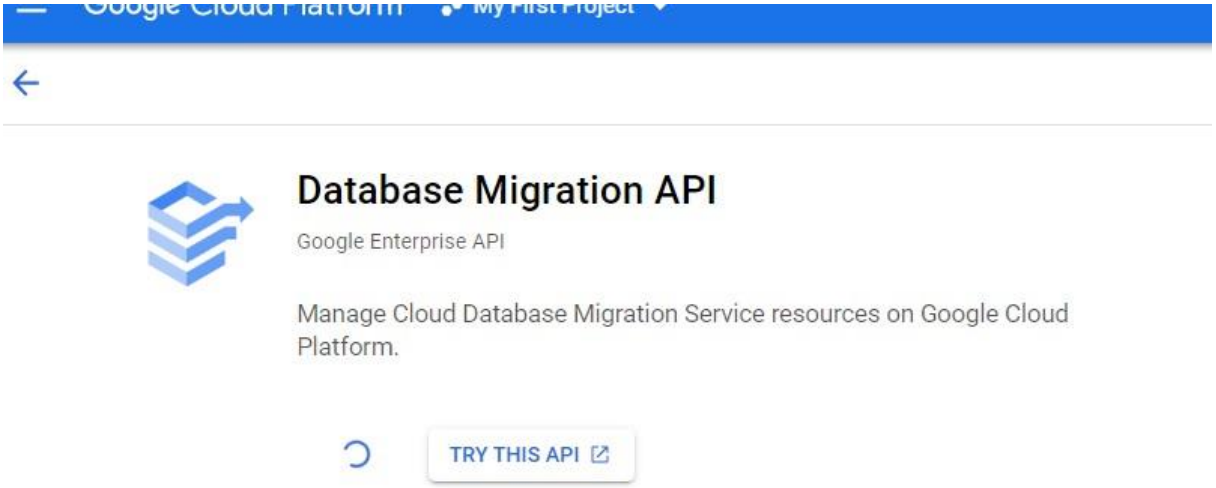
Create a Bucket and upload the given db file.



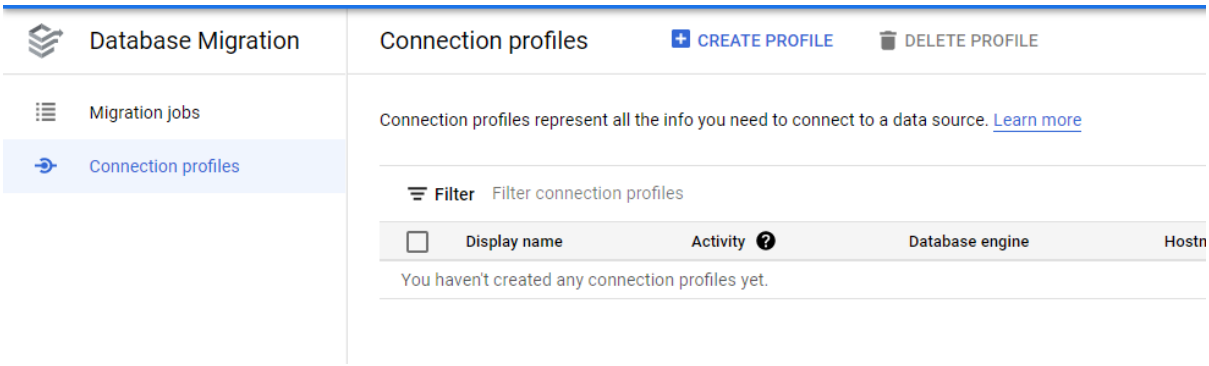
Go to SQL and create an instance.

And import the file from bucket.

Now,Go to Database Migration and Enable the below API.

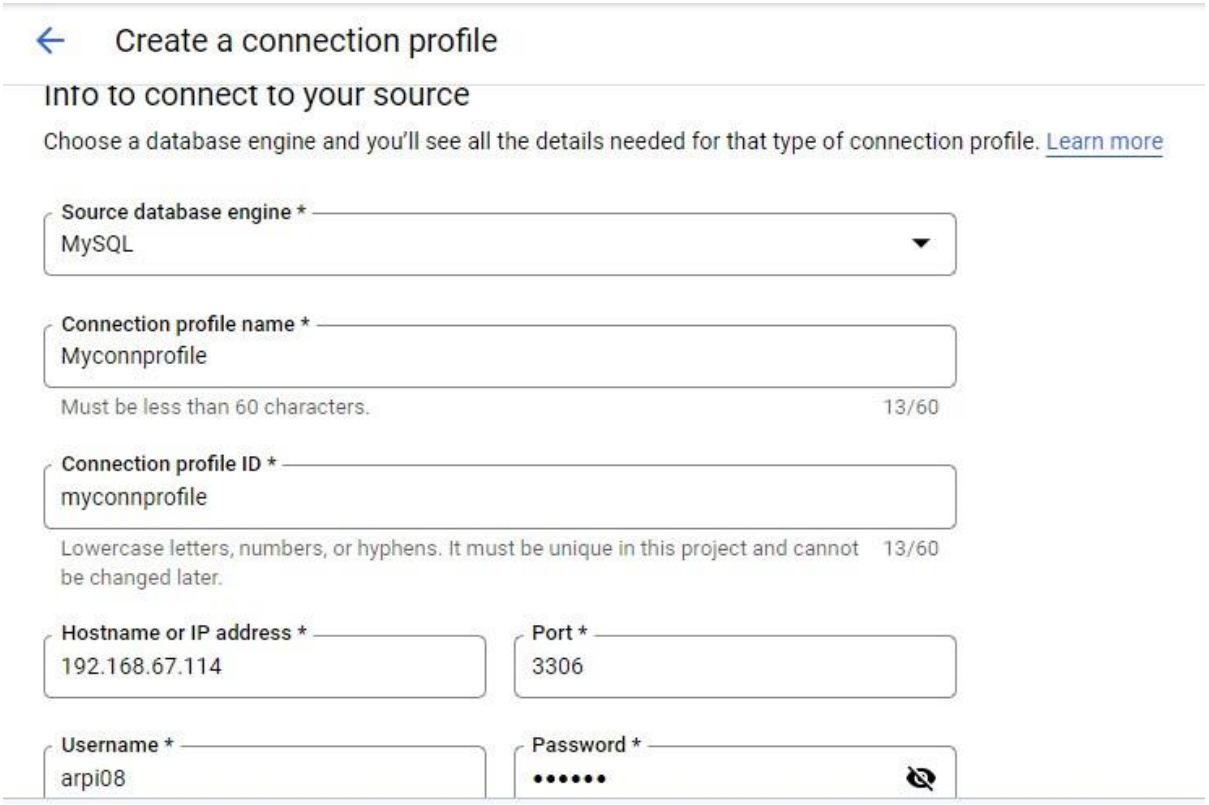


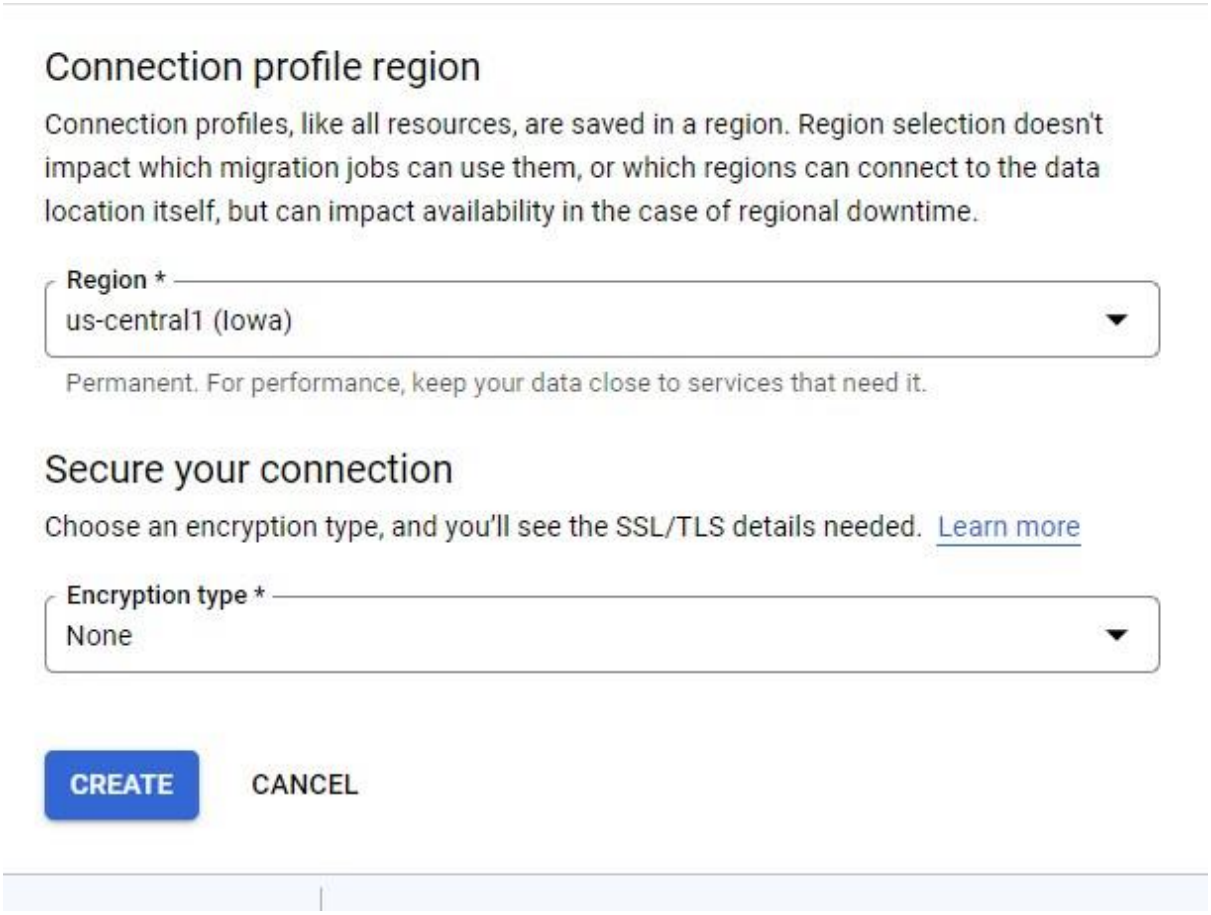
Create a connection profile.



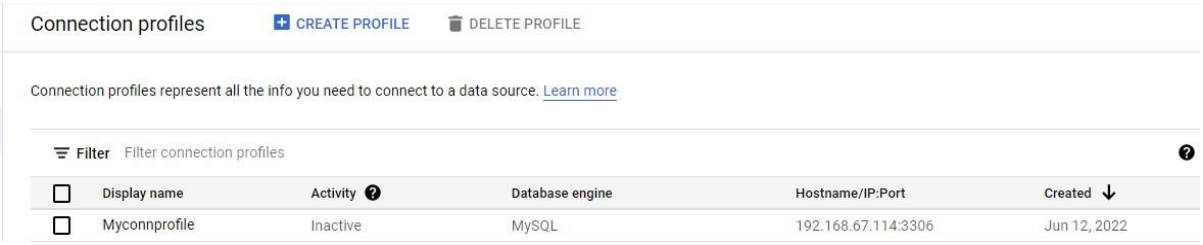
select MySQL

Enter details.

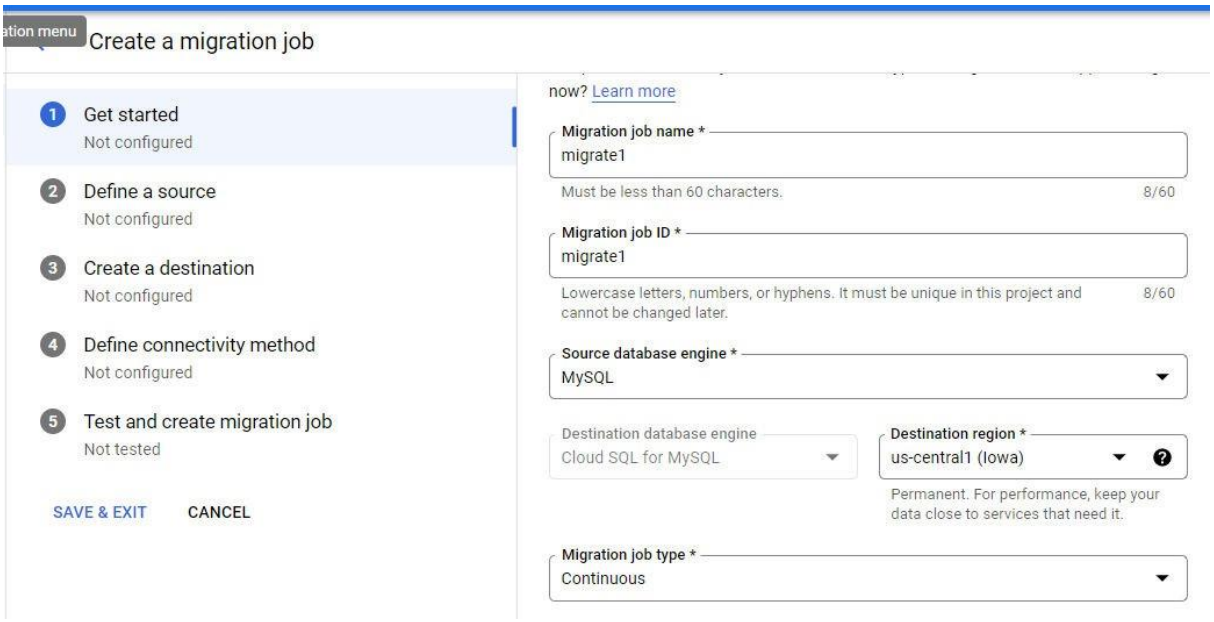




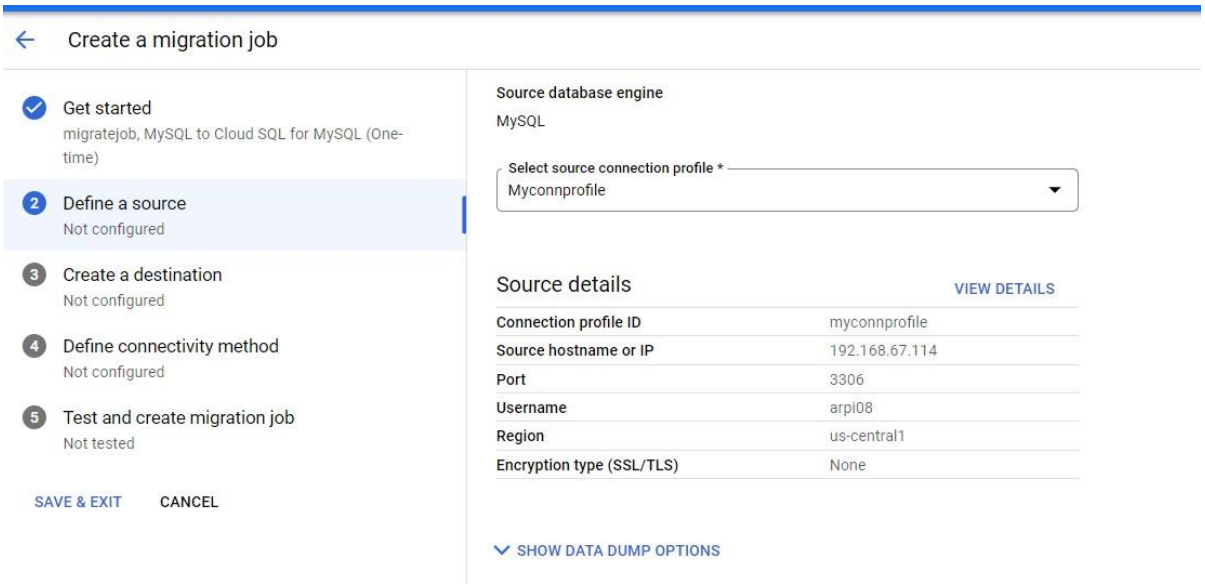
Connection profile created.



Now, Create a migration job.



Define source.



Create a destination

A new destination instance will be created for your migration job. Provide some basic info for it, and configure as needed. Learn more

Destination database engine

Cloud SQL for MySQL

Destination Instance ID *

migratejob

Use lowercase letters, numbers, and hyphens. Start with a letter. Note that the ID can't be changed later.

Password *

••••••••••••••••

GENERATE

Set a password for the default admin user "root". Learn more

Database version *

Cloud SQL for MySQL 5.7

Can be at most one version above the source version

Database version *

Cloud SQL for MySQL 5.7

Can be at most one version above the source version

Choose region and zone

For better performance, keep your data close to the services that need it. Region is permanent, while zone can be changed any time.

Region

us-central1 (Iowa)

Zone *

us-central1-c

Configure your destination instance

You can customize the new instance associated with this migration job. Choices affect billing according to Cloud SQL Pricing .

Connections

Choose a network path for connecting to this instance. For extra security, consider

Machine Type

Choose a preset or customize your own. For better performance, choose a machine type with enough memory to hold your largest table.

High memory

- (•) 4 vCPU, 26 GB
- ( ) 8 vCPU, 52 GB
- ( ) 16 vCPU, 104 GB
- ( ) Custom

Storage

Storage type

Choice is permanent. Storage type affects performance.

- (•) SSD (Recommended)
  Most popular choice. Lower latency than HDD with higher QPS and data throughput.
- ( ) HDD

If you plan on migrating via IP allowlist, then select the Public IP check box. If you plan on migrating via VPC-peering or Reverse SSH tunnel, then select the Private IP check box.

- [ ] Private IP

Requires additional APIs and permissions, which may require your system admins. Can't be disabled once enabled. Learn more

The Service Networking API must be enabled in order to enable Private IP for this instance.

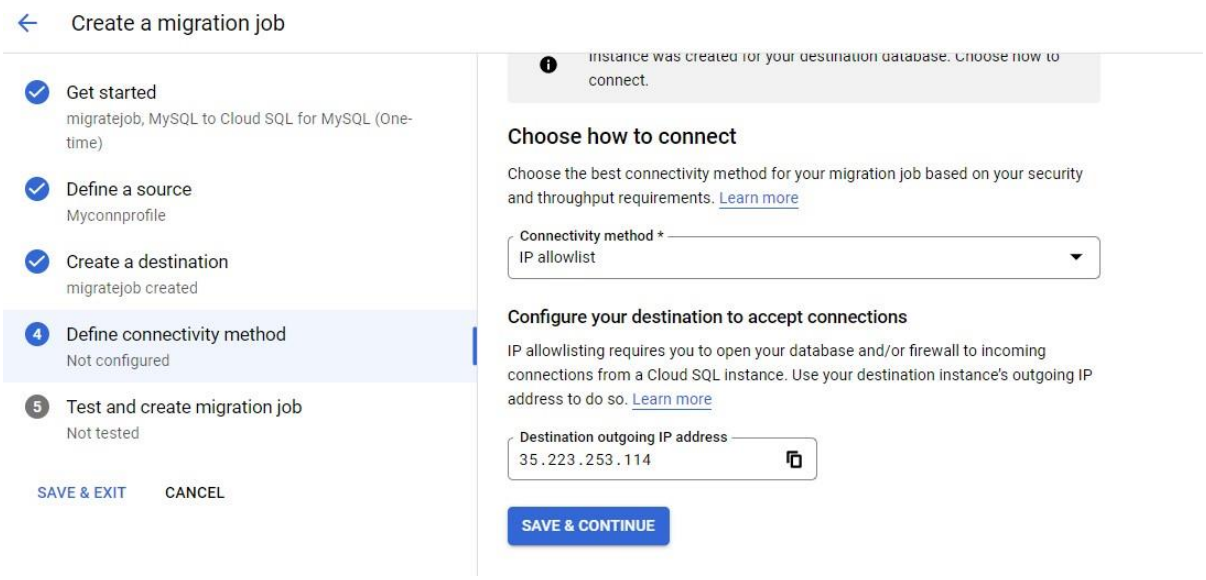
RETRY

- [x] Public IP

Authorize a network or use Cloud SQL Proxy to connect to this instance. Learn more

Authorized networks

ADD NETWORK



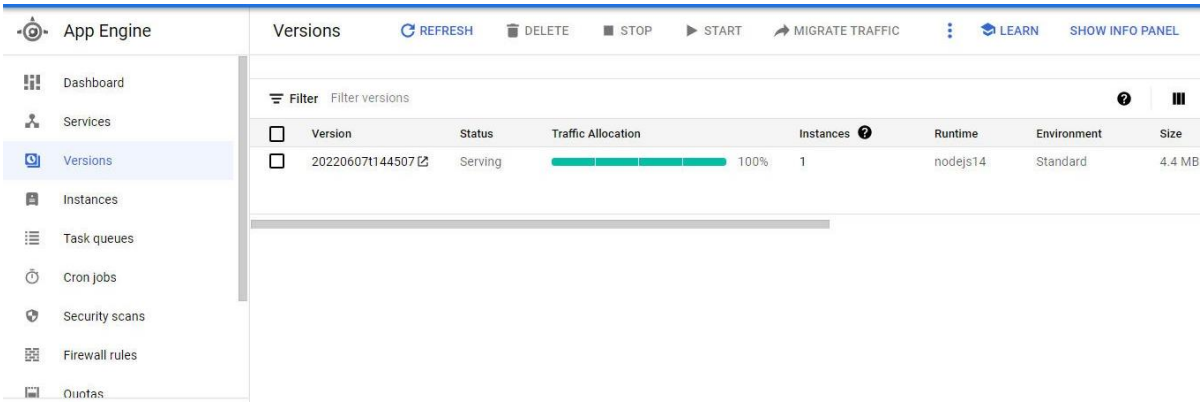
Test and create. Migration done successfully.

Check in Cloud SQL. File will be available.

4. The company wants to review different versions of the code on the app service, explain the mechanism using screenshots.

Go to App Engine> Versions.

On console, It is created as ver1.



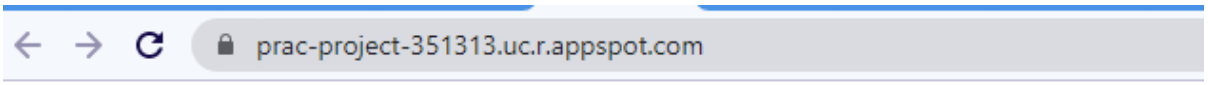
Edit the app.js file as Hello world from intellipaat.

Deploy as ver2.

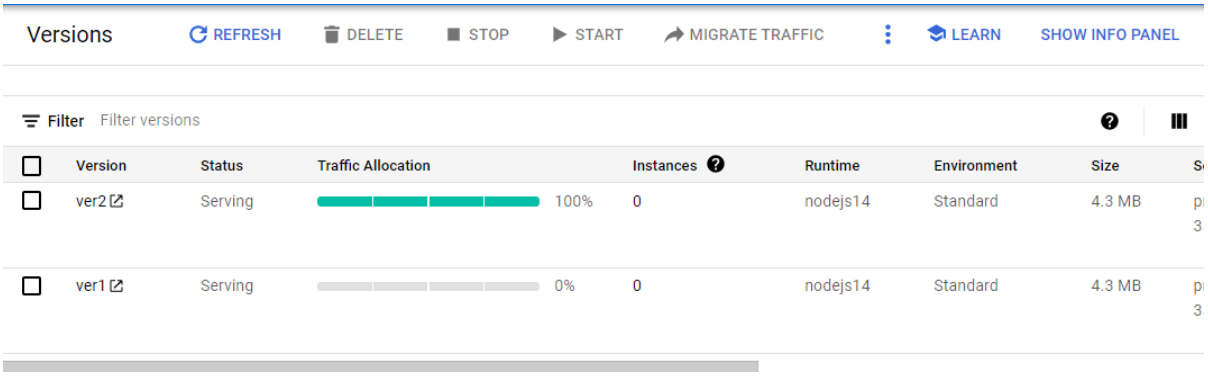
```
app.js  app.yaml  node_modules  package.json  package-lock.json  README.md  test
arpitacs0024@cloudshell:~/cloudshell_open/nodejs-docs-samples-5/appengine/hello-world/standard (prac-project-351313)$
arpitacs0024@cloudshell:~/cloudshell_open/nodejs-docs-samples-5/appengine/hello-world/standard (prac-project-351313)$ vi app.js
arpitacs0024@cloudshell:~/cloudshell_open/nodejs-docs-samples-5/appengine/hello-world/standard (prac-project-351313)$ gcloud app deploy -v ver2
Services to deploy:

descriptor:                  [/home/arpitacs0024/cloudshell_open/nodejs-docs-samples-5/appengine/hello-world/standard/app.yaml]
source:                      [/home/arpitacs0024/cloudshell_open/nodejs-docs-samples-5/appengine/hello-world/standard]
target project:              [prac-project-351313]
target service:              [default]
target version:              [ver2]
target url:                  [https://prac-project-351313.uc.r.appspot.com]
target service account:      [App Engine default service account]


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

Beginning deployment of service [default]...
```

Browse it.

```
arpitacs0024@cloudshell:~/cloudshell_open/nodejs-docs-samples-5/appengine/hello-world/standard (prac-project-351313)$ gcloud app browse
Did not detect your browser. Go to this link to view your app:
https://prac-project-351313.uc.r.appspot.com
arpitacs0024@cloudshell:~/cloudshell_open/nodejs-docs-samples-5/appengine/hello-world/standard (prac-project-351313)$
```



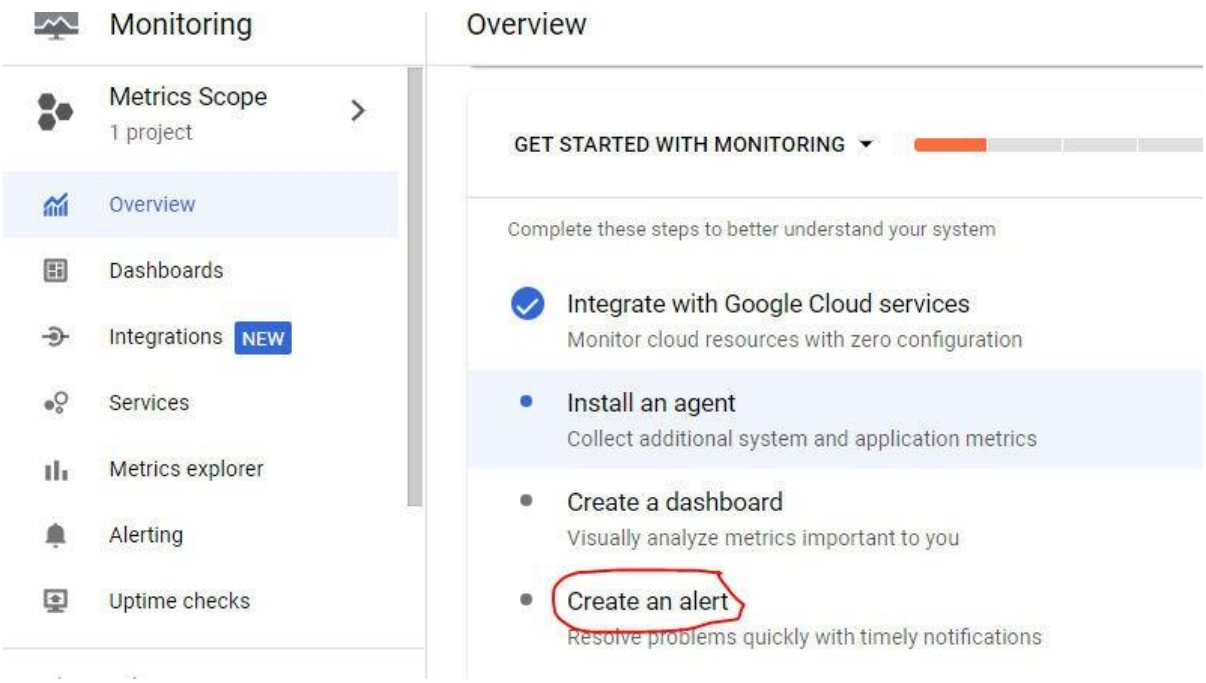
Hello, world from intellipaat



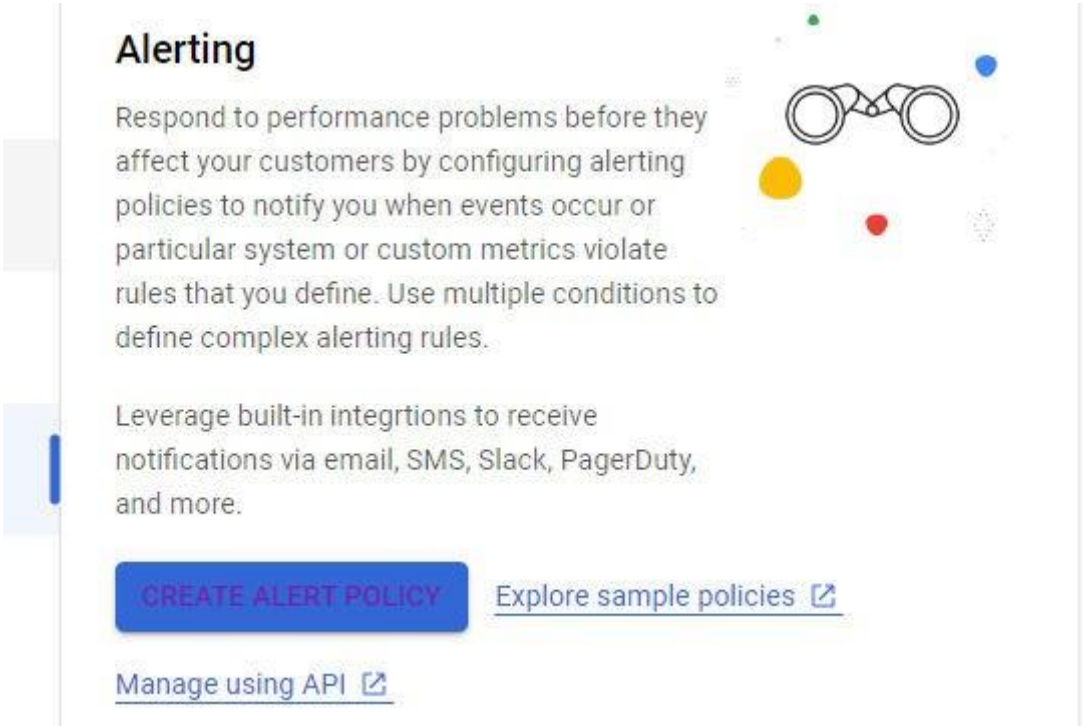
5. Configure Stackdriver monitoring to monitor the storage server, and email should be triggered to the owner of the project in case this server goes down.

Go to stackdriver > Monitoring

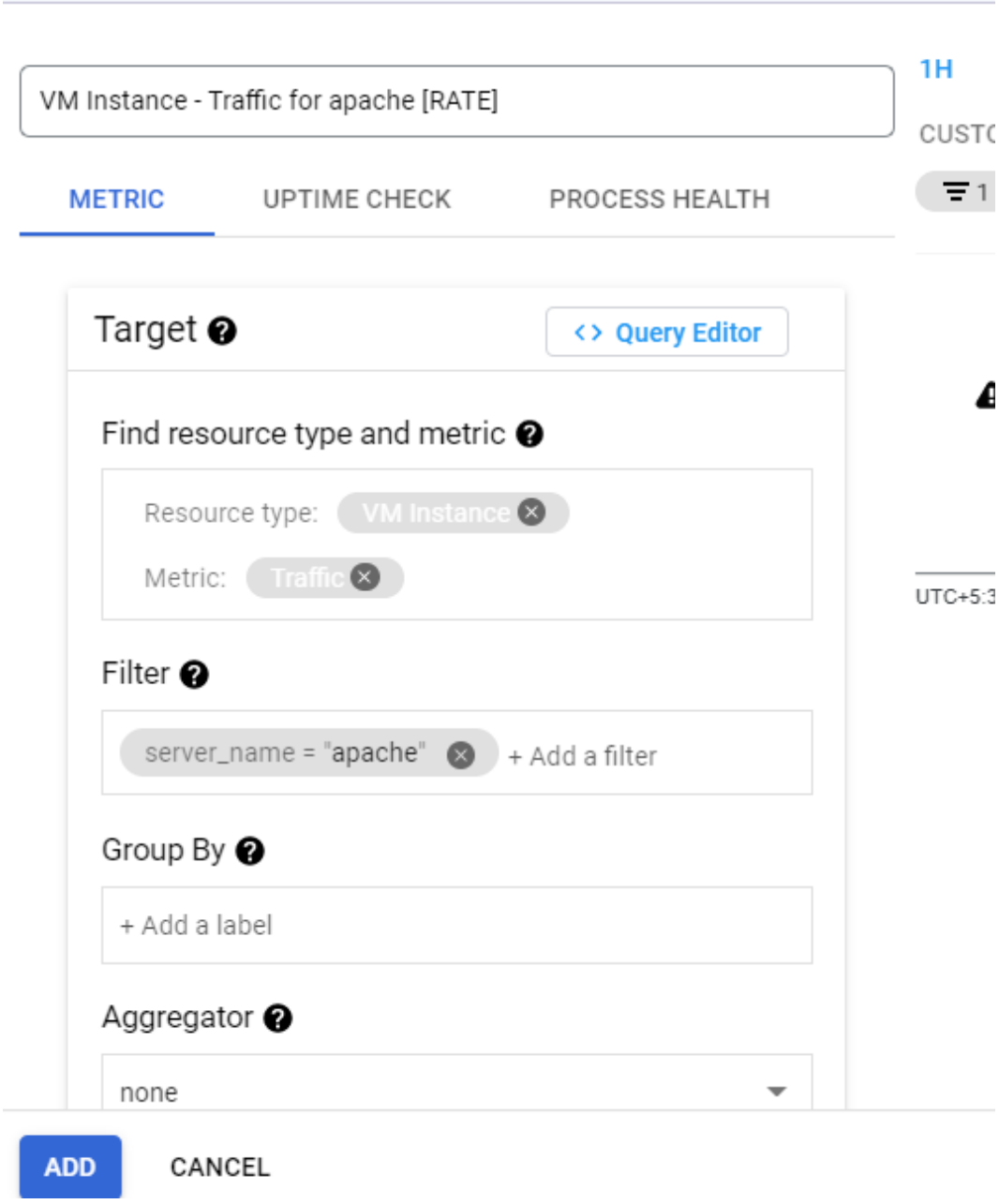
Create a alert.

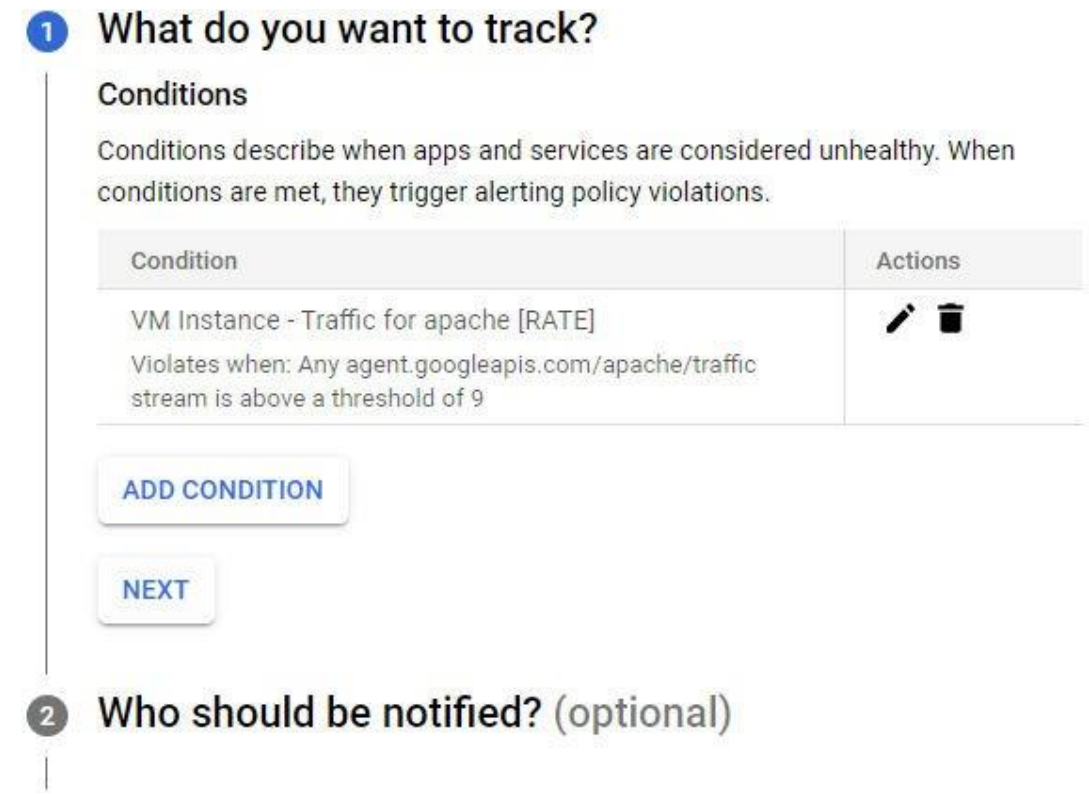


Now, Create a alert policy.

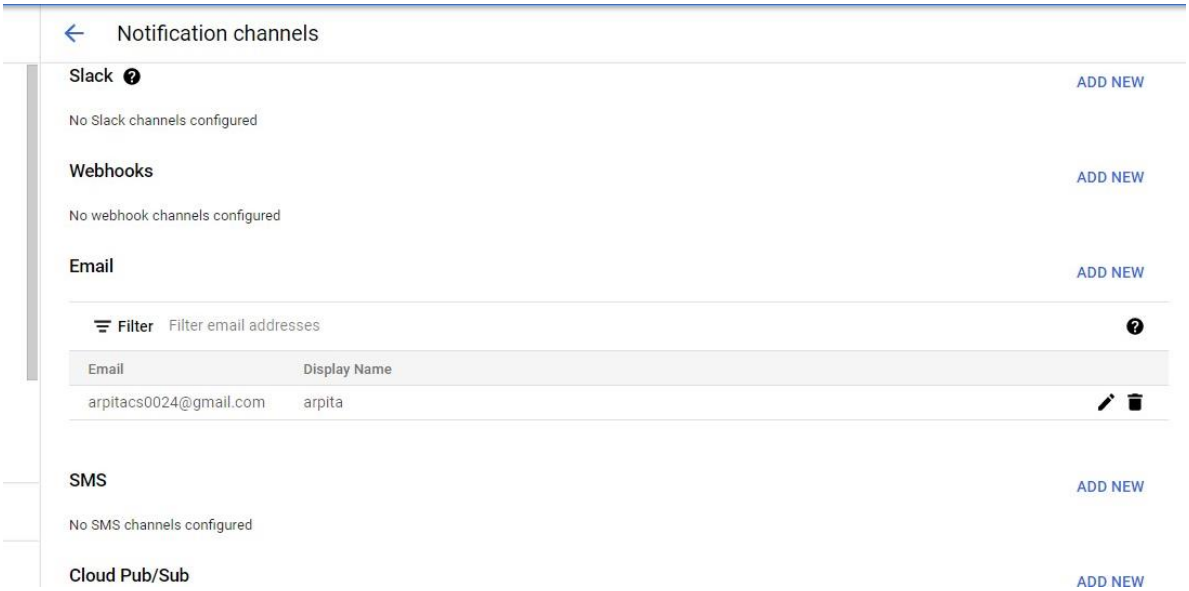


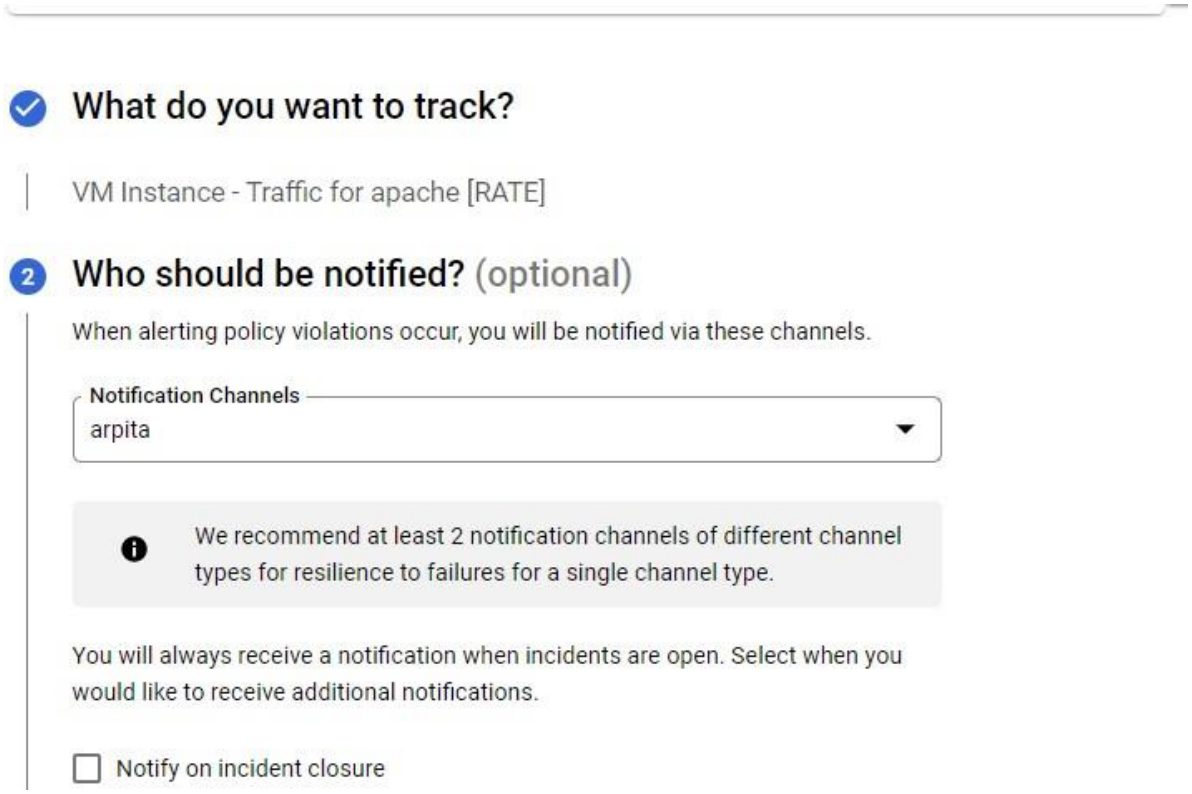
Select metric VM instance. Select metric as uptime.


VM Instance - Traffic for apache [RATE]
1H
CUSTO
1
METRIC
UPTIME CHECK
PROCESS HEALTH
Target
Query Editor
Find resource type and metric
Resource type:
VM Instance
Metric:
Traffic
UTC+5:3
Filter
server_name = "apache"
+ Add a filter
Group By
+ Add a label
Aggregator
none
ADD
CANCEL

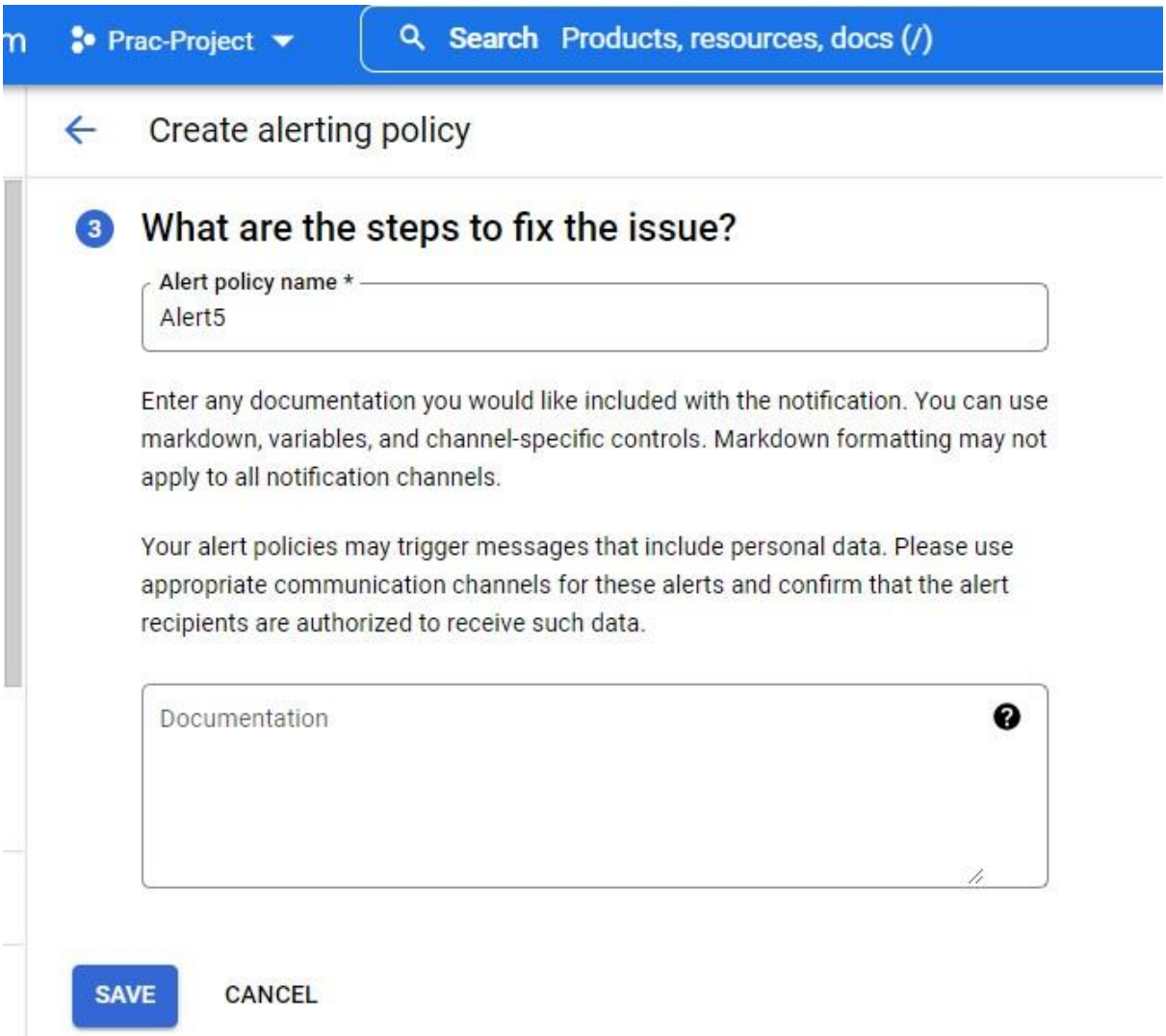
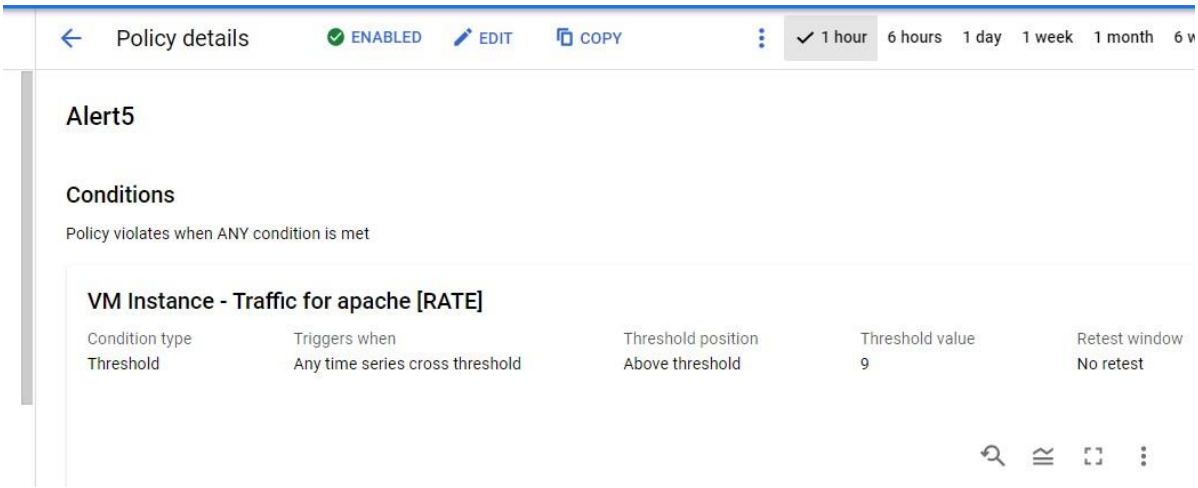


Add notification channel and add email.





Name the alert.



Policy created for apache service.