

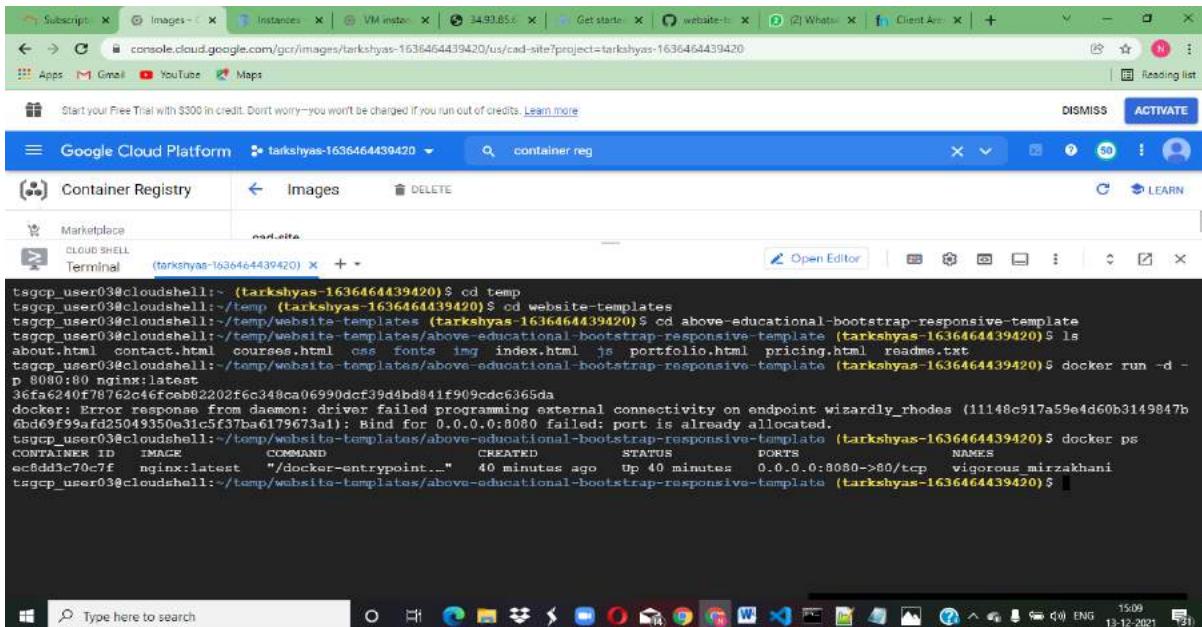
# GOOGLE SRE TRAINING

## FINAL ASSESSMENT

### 1. Create a Linux VM and host a sample website given in below link

<https://github.com/learning-zone/website-templates/tree/master/above-educational-bootstrap-responsive-templates>

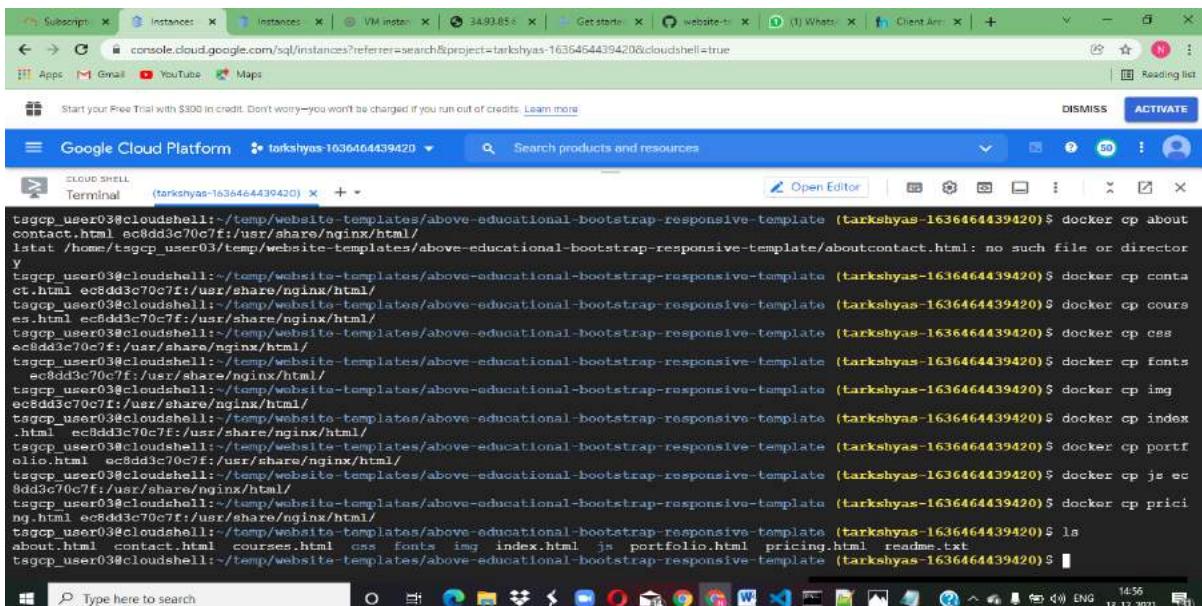
To host this sample website, first push the website contents in to container registry as follows:



The screenshot shows a Google Cloud Platform interface. In the top navigation bar, the URL is `console.cloud.google.com/gcr/images/tarkshyas-1636464439420/us/cad-site?project=tarkshyas-1636464439420`. Below the navigation bar, there's a banner for a free trial with \$300 in credit. The main area is titled "Container Registry" and shows a "Marketplace" tab selected. A terminal window is open with the following command history:

```
tsgcp_user03@cloudshell:~ (tarkshyas-1636464439420)$ cd temp
tsgcp_user03@cloudshell:~/temp (tarkshyas-1636464439420)$ cd website-templates
tsgcp_user03@cloudshell:~/temp/website-templates (tarkshyas-1636464439420)$ cd above-educational-bootstrap-responsive-template
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$ ls
about.html contact.html courses.html css fonts img index.html js portfolio.html pricing.html readme.txt
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$ docker run -d -p 8090:80 nginx:latest
36fa6240f78762c46fcab02202f6cc348ca06990dcf39ddbd841f909cd6365da
docker: Error response from daemon: driver failed programming external connectivity on endpoint wizardly_rhodes (11140c917a59e4d60b3149847b6b69f99afdf25049350e31c5f37ba6179673a1): Bind for 0.0.0.0:8090 failed: port is already allocated.
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
ecdd3c70c7f nginx:latest "/docker-entrypoint..." 40 minutes ago Up 40 minutes 0.0.0.0:8090>80/tcp vigorous_mirzakhani
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$
```

Copying all the files to container.



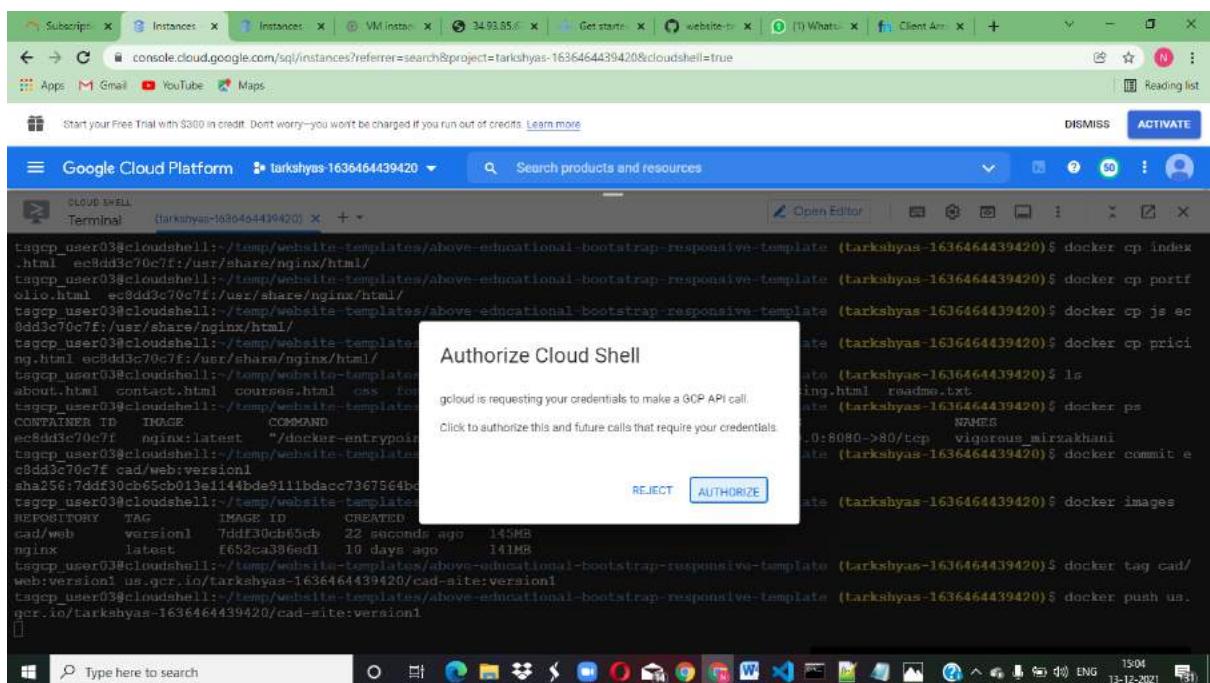
The screenshot shows a Google Cloud Platform interface. In the top navigation bar, the URL is `console.cloud.google.com/sql/instances?referter=search&project=tarkshyas-1636464439420&cloudshell=true`. Below the navigation bar, there's a banner for a free trial with \$300 in credit. The main area is titled "Cloud Shell" and shows a terminal window with the following command history:

```
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$ docker cp about contact.html ec8dd3c70c7f:/usr/share/nginx/html/
lsat /home/tsgcp_user03/temp/website-templates/above-educational-bootstrap-responsive-template/aboutcontact.html: no such file or directory
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$ docker cp conta ct.html ec8dd3c70c7f:/usr/share/nginx/html/
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$ docker cp cours es.html ec8dd3c70c7f:/usr/share/nginx/html/
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$ docker cp css ec8dd3c70c7f:/usr/share/nginx/html/
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$ docker cp fonts ec8dd3c70c7f:/usr/share/nginx/html/
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$ docker cp img ec8dd3c70c7f:/usr/share/nginx/html/
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$ docker cp index.html ec8dd3c70c7f:/usr/share/nginx/html/
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$ docker cp portfolio.html ec8dd3c70c7f:/usr/share/nginx/html/
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$ docker cp pricing.html ec8dd3c70c7f:/usr/share/nginx/html/
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$ ls
about.html contact.html courses.html css fonts img index.html js portfolio.html pricing.html readme.txt
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$
```

Tag the docker with zone and region.

```
takshyaas-1636464439420$ docker cp fonts
ec8dd3c70c7f:/usr/share/nginx/html/
takshyaas-1636464439420$ docker cp img
ec8dd3c70c7f:/usr/share/nginx/html/
takshyaas-1636464439420$ docker cp index
.html ec8dd3c70c7f:/usr/share/nginx/html/
takshyaas-1636464439420$ docker cp portf
olio.html ec8dd3c70c7f:/usr/share/nginx/html/
takshyaas-1636464439420$ docker cp portfolio.html
takshyaas-1636464439420$ docker cp js ec
8dd3c70c7f:/usr/share/nginx/html/
takshyaas-1636464439420$ docker cp pri
cing.html ec8dd3c70c7f:/usr/share/nginx/html/
takshyaas-1636464439420$ ls
about.html contact.html courses.html css fonts img index.html js portfolio.html pricing.html readme.txt
takshyaas-1636464439420$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
ec8dd3c70c7f nginx:latest "/docker-entrypoint..." 30 minutes ago Up 30 minutes 0.0.0.0:8080->80/tcp vigorous_mirzakhani
takshyaas-1636464439420$ docker commit e
c8dd3c70c7f cad/web:version1
sha256:7ddf30cb65cb013e1144bde911bdacc7367564bde89abb8ffad4b46c62e96
takshyaas-1636464439420$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
cad/web version1 7ddf30cb65cb 22 seconds ago 145MB
nginx latest f652ca386ed1 10 days ago 141MB
takshyaas-1636464439420$
```

Push the image to container registry.



```

about.html contact.html courses.html css fonts img index.html js portfolio.html pricing.html readme.txt
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
ec8dd3c70c7f nginx:latest "/docker-entrypoint..." 30 minutes ago Up 30 minutes 0.0.0.0:8080->80/tcp vigorous_mirzakhani
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$ docker commit ec8dd3c70c7f cad/web:version1
sha256:7ddf30cb65cb013e1144bde9111bdacc7367564bde89abbc0fia48b46c62e96
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
cad/web version1 7ddf30cb65cb 22 seconds ago 145MB
nginx latest f652ca386ed1 10 days ago 141MB
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$ docker tag cad/web:version1 us.gcr.io/tarkshyas-1636464439420/cad-site:version1
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$ docker push us.gcr.io/tarkshyas-1636464439420/cad-site:version1
The push refers to repository us.gcr.io/tarkshyas-1636464439420/cad-site
1000e5942652: pushed
2bedd47a66c07: Layer already exists
82caad489ad7: Layer already exists
d3eldca44e82: Layer already exists
c9ffcd9cc0d8: Layer already exists
0664b7821b60: Layer already exists
9321ff862abb: Layer already exists
version1: digest: sha256:34ec33654c5c77705179dd25cc8321d7b91c52ac475c86b6f4aa080cd09545e0 size: 1781
tsgcp_user03@cloudshell:~/temp/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1636464439420)$

```

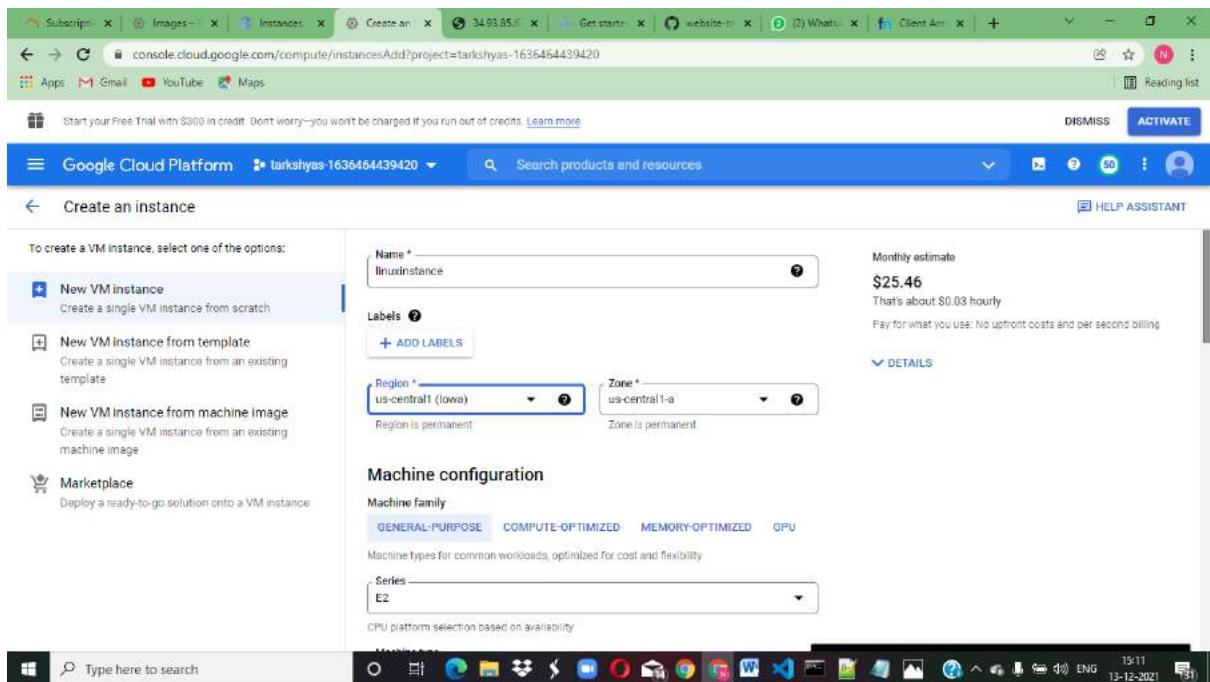
Now the image is in Container Repository.

Name	Tags	Virtual Size	Created	Uploaded	Vulnerabilities
34ec33654c5c	version1	56.3 MB	6 minutes ago	1 minute ago	Never scanned
90d350d088d8		54.1 MB	2 hours ago	2 hours ago	Never scanned

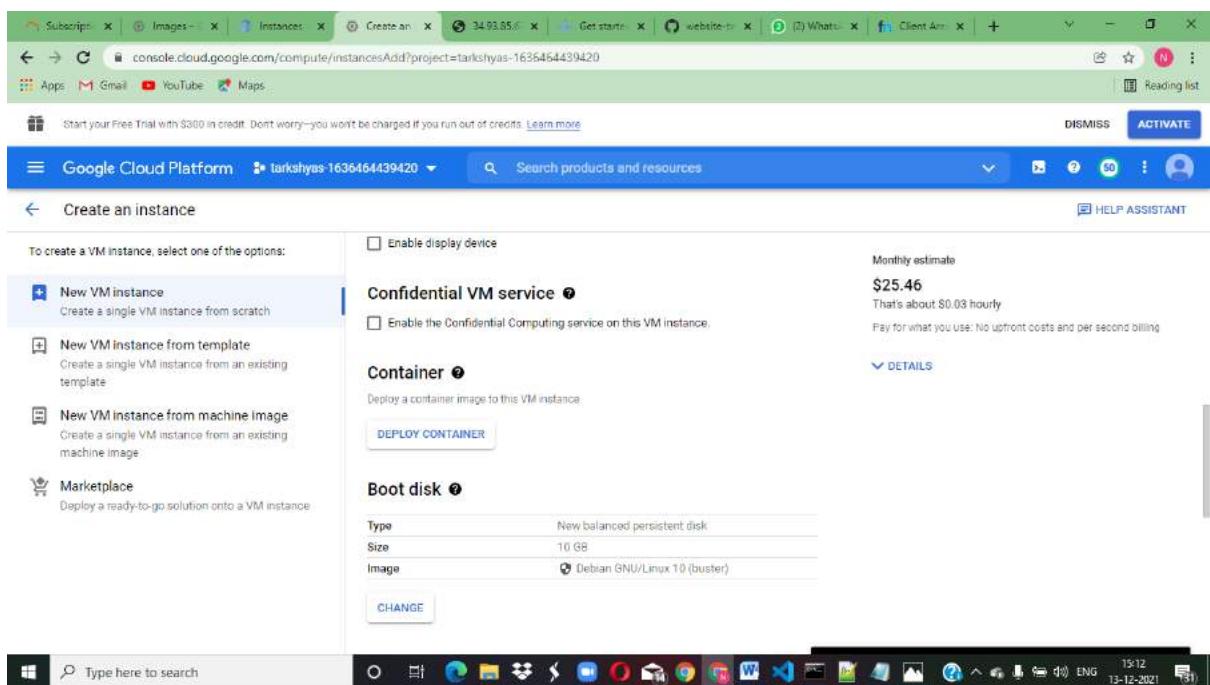
Now create Linux VM instance :

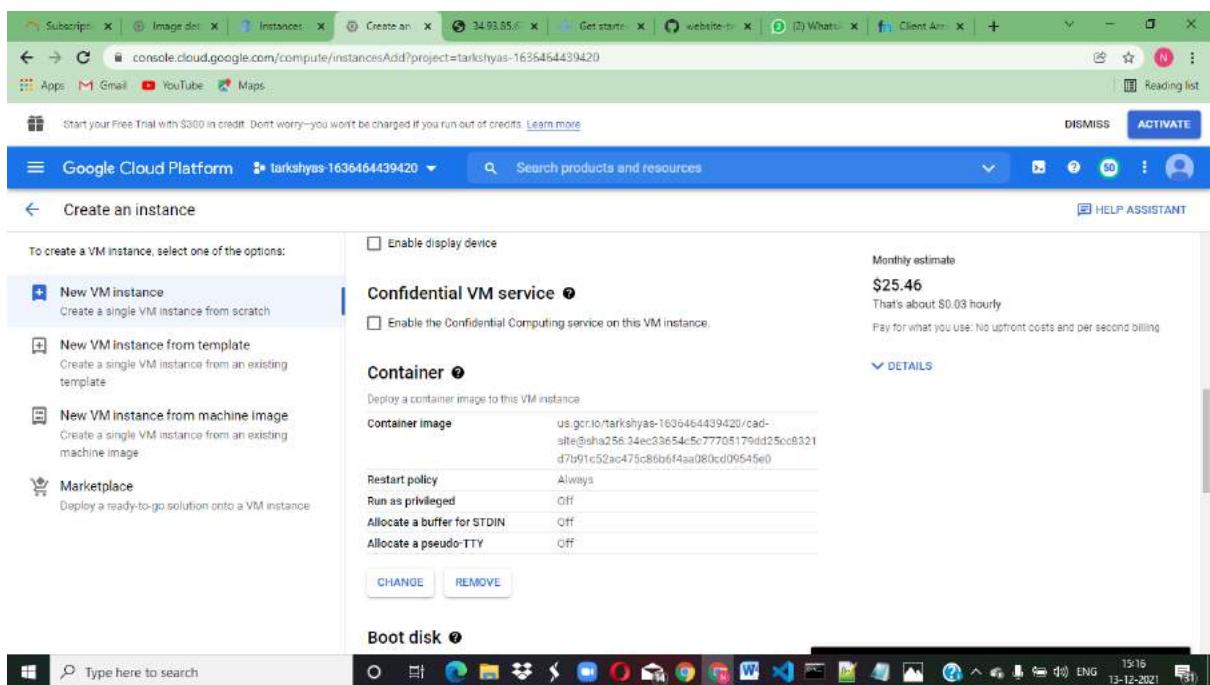
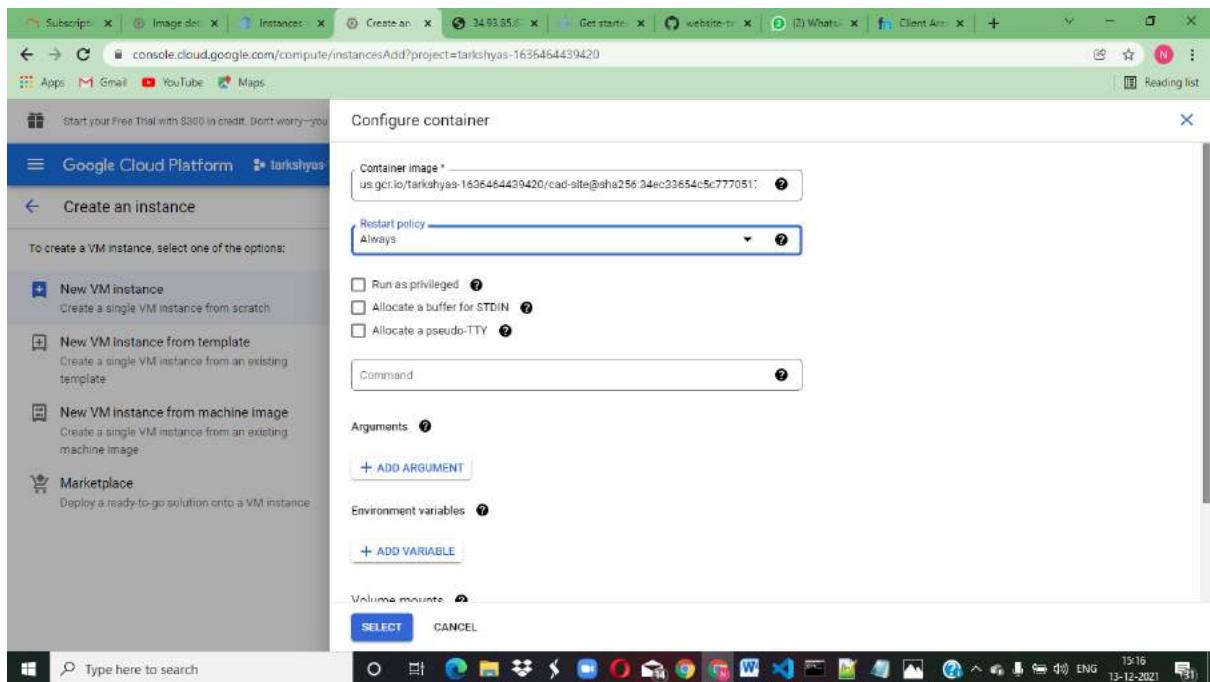
For creating VM, Click on compute engine -> VM instances

Click create an instance and specify name and region for your VM.



Specify container image name and optional configuration parameters.

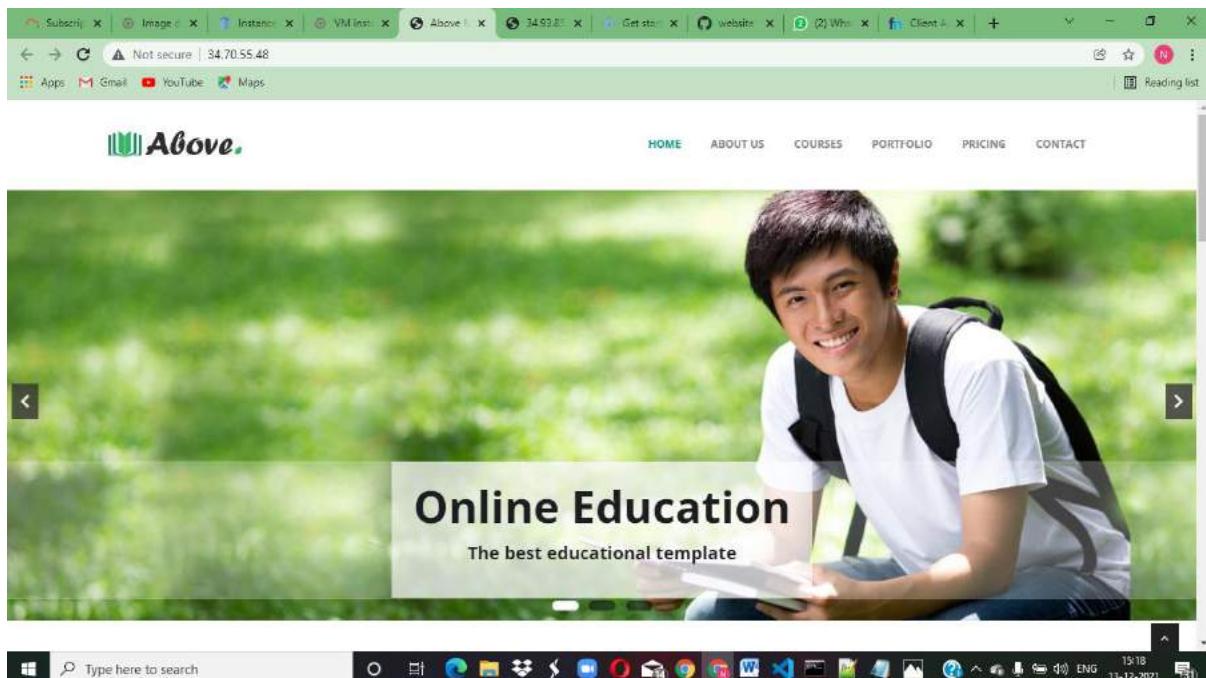




The screenshot shows the 'Create an instance' wizard in the Google Cloud Platform. The left sidebar lists options: 'New VM instance', 'New VM instance from template', 'New VM instance from machine image', and 'Marketplace'. The main area shows 'Compute Engine: Us-central1-a' selected. Under 'Access scopes', 'Allow default access' is chosen. A monthly estimate of '\$25.46' is displayed. Under 'Firewall', 'Allow HTTP traffic' is checked. A note states: 'You will be billed for this instance. Compute Engine pricing'. At the bottom are 'CREATE', 'CANCEL', and 'EQUIVALENT COMMAND LINE' buttons.

Now VM instance is created. Click on External IP to view the sample website.

The screenshot shows the 'VM instances' page in the Google Cloud Platform. The left sidebar is under 'Compute Engine' and includes 'Virtual machines' (selected), 'VM instances' (highlighted), 'Instance templates', 'Sole-tenant nodes', 'Machine images', 'TPUs', 'Committed use discounts', 'Migrate for Compute Eng...', 'Marketplace', and 'Release Notes'. The main area displays a table of VM instances. One row is selected, showing 'Status' as 'Running', 'Name' as 'linuxinstance', and 'Zone' as 'us-central1-a'. To the right, a 'Select an instance' panel shows a message: 'Please select at least one resource.' A sidebar on the right provides 'Start your project' tips, 'Find existing VM solutions', 'How-to guides and tutorials', and 'Explore tools and APIs'.

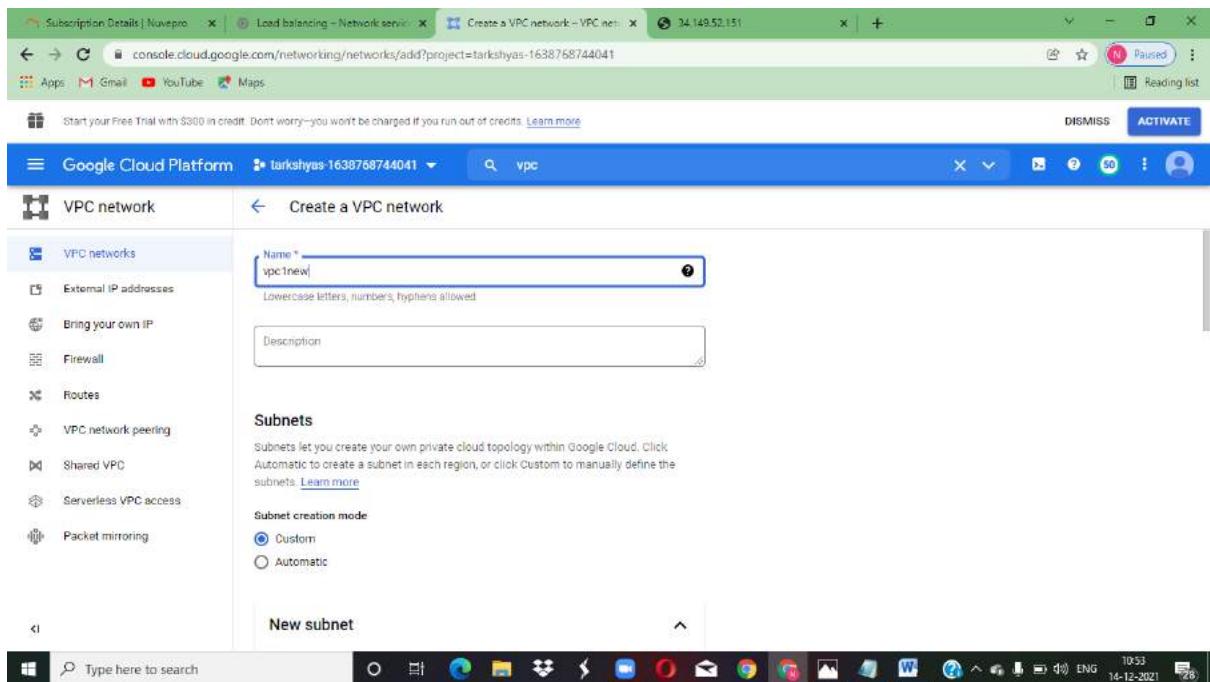


## 2. Create a VPC Network to communicate between 2 VM instance in different networks.

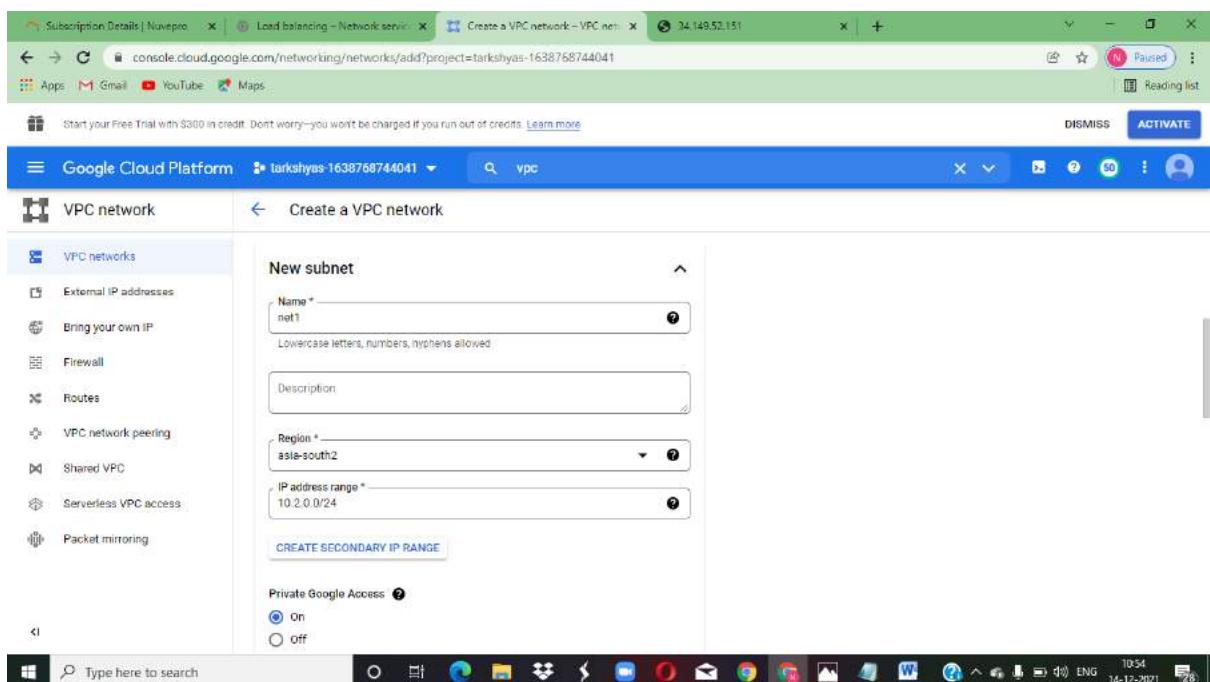
Go to to VPC Network. Click on create VPC Network.

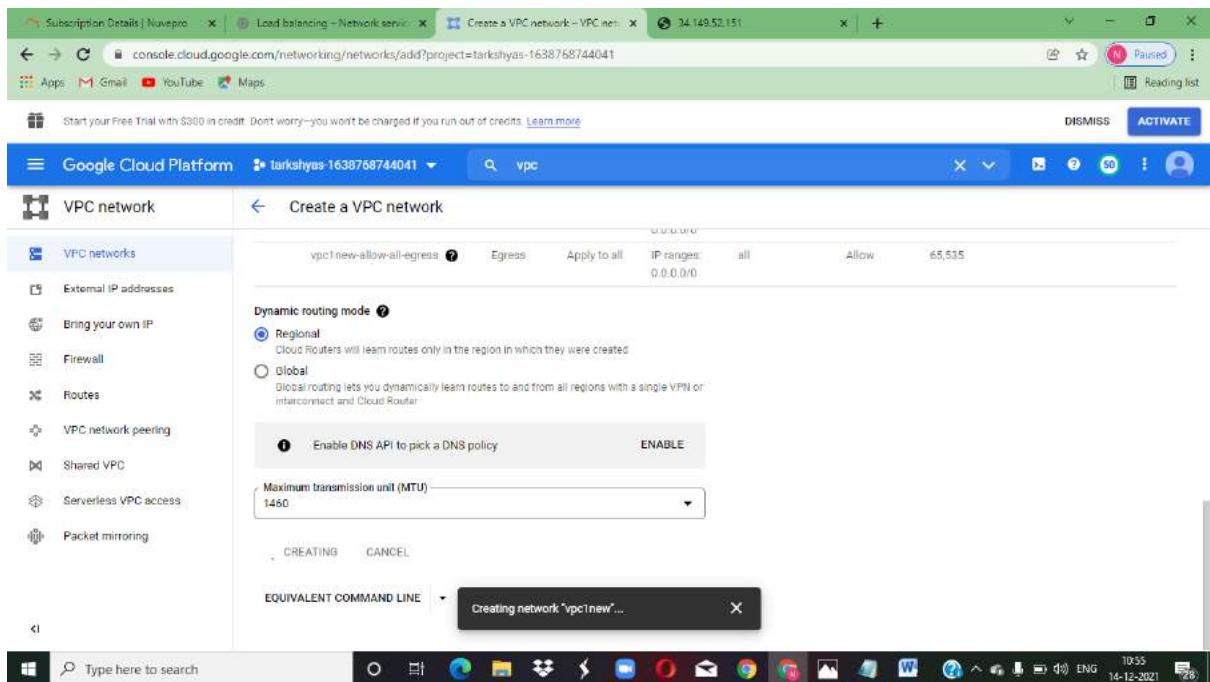
Name	Region	Subnets	MTU	Mode	IP address ranges	Gateways	Firewall Rules	Global dynamic routing	Flow
vpc1new	us-central1	29	1460	Auto	10.128.0.0/20	10.128.0.1	5	Off	Off

Name the VPC Network as vpc1new and choose custom for the subset creation mode.



Specify configuration parameters for a subset as follows:

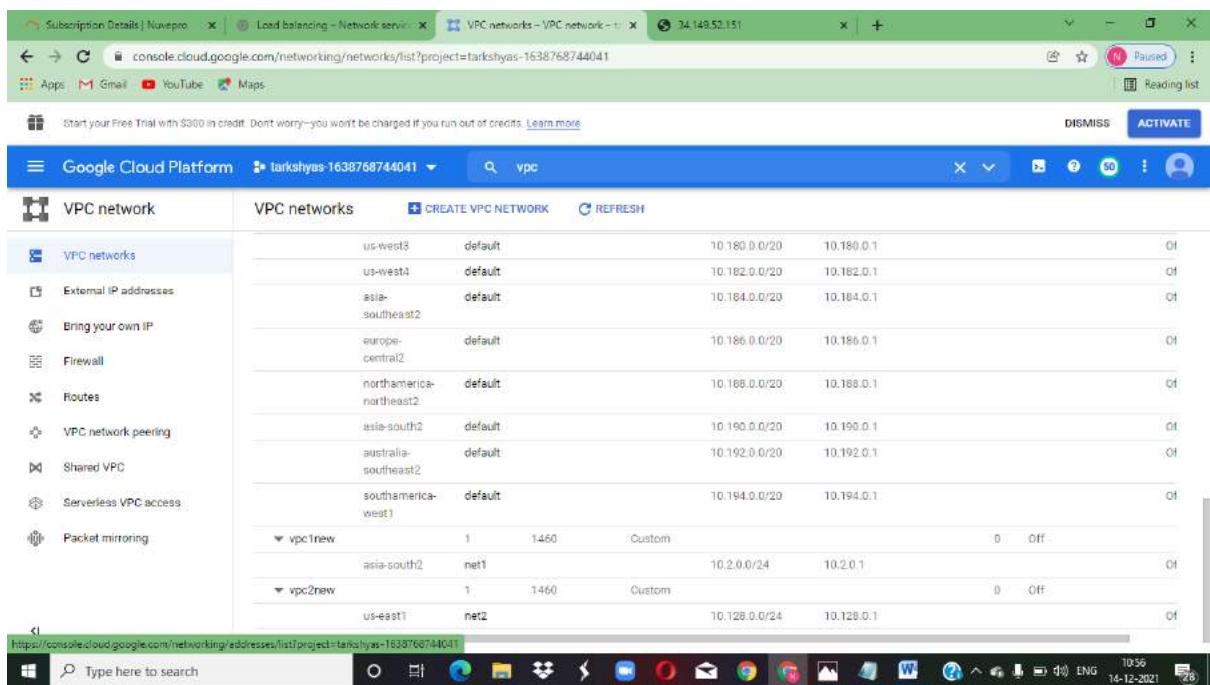




Click create.

Likewise create another VPC Network named `vpc2new`.

Thus two VPC Networks are created.



Reserve 2 external IP addresses for these two VPC Networks.

The screenshot shows the Google Cloud Platform interface for managing VPC networks. On the left, a sidebar lists various network-related services: VPC networks, External IP addresses, Bring your own IP, Firewall, Routes, VPC network peering, Shared VPC, Serverless VPC access, and Packet mirroring. The 'External IP addresses' section is selected. In the main pane, a table lists existing static IP addresses, including 'ib-ipv4-1' (34.149.52.151), 'new-ip' (34.122.51.89), 'vpc1ext' (24.131.19.32), 'vpc2ext' (35.229.104.134), and others. A modal window at the bottom right indicates that a new address has been successfully created: 'Successfully created address "vpc1ext"'.

For creating Firewall rules click on firewall -> Create firewall rule.

Name the firewall and network.

The screenshot shows the Google Cloud Platform interface for creating a new firewall rule. The sidebar on the left is identical to the previous screenshot, with 'Firewall' selected. The main pane displays a form for creating a 'Create a firewall rule'. The 'Name' field is set to 'vpcf1'. The 'Logs' section is turned off ('off'). The 'Network' dropdown is set to 'vpc1new'. The 'Priority' is set to '1000'. A note at the bottom states 'Priority can be 0 - 65535'.

The screenshot shows the Google Cloud Platform interface for creating a firewall rule. The left sidebar is titled 'VPC network' and lists options like 'VPC networks', 'External IP addresses', 'Firewall' (which is selected), 'Routes', 'VPC network peering', 'Shared VPC', 'Serverless VPC access', and 'Packet mirroring'. The main panel is titled 'Create a firewall rule' and contains the following fields:

- Source filter:** IPv4 ranges (dropdown set to 'IPv4 ranges')  
Source IPv4 ranges: 0.0.0.0/0
- Second source filter:** None
- Protocols and ports:** Allow all (radio button selected)
- DISABLE RULE:** (checkbox)
- CREATE** (button) | CANCEL (button)

Below the form, there's a section for 'EQUIVALENT COMMAND LINE'.

First firewall rule is created as vpcr1.

Create firewall rule vpcr2 .

The screenshot shows the Google Cloud Platform interface for managing VPC firewall rules. The left sidebar is identical to the previous screenshot. The main panel is titled 'Firewall' and displays a table of existing rules:

Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network	Logs
rule1	Ingress	allow-health	IP ranges: 130.1	Tcp:80	Allow	1000	default	Off
default-allow-icmp	Ingress	Apply to all	IP ranges: 0.0.0	ICMP	Allow	65534	default	Off
default-allow-internal	Ingress	Apply to all	IP ranges: 10.12	Tcp:0-65535 Udp:0-65535 ICMP	Allow	65534	default	Off
default-allow-rdp	Ingress	Apply to all	IP ranges: 0.0.0	Tcp:3389	Allow	65534	default	Off
default-allow-ssh	Ingress	Apply to all	IP ranges: 0.0.0	Tcp:22	Allow	65534	default	Off
vpcr1	Ingress	Apply to all	IP ranges: 0.0.0	All	Allow	1000	vpc1new	Off
vpcr2	Ingress	Apply to all	IP ranges: 0.0.0	All	Allow	1000	vpc2new	Off

Create two instances belongs to these two VPC Networks.

Here it is named as vpc1instance and vpc2instance.

The screenshot shows the Google Cloud Platform Compute Engine interface. On the left, a sidebar lists options like VM instances, Instance templates, Sole-tenant nodes, Machine images, TPUs, Committed use discounts, Migrate for Compute Engine, Marketplace, and Release Notes. The main area displays a table of VM instances:

Status	Name	Zone
Green	instancegrp1-7mm	us-central1-a
Green	instancegrp1-cv3z	us-central1-a
Green	instancegrp1-gcpc	us-central1-a
Green	vpc1instance	asia-south2-a
Green	vpc2instance	us-east1-b

A modal window titled "Select an instance" is open, showing tabs for PERMISSIONS, LABELS, and MONITORING. It contains the message: "Please select at least one resource." To the right of the main interface, there's a sidebar titled "Choose the right storage for your project" with links to connecting instances to Cloud Storage buckets, Cloud SQL databases, and adding disk space. Below that is a section titled "Find ways to save money" with links to preemptible instances, committed use discounts, custom machine types, and applying sizing recommendations. At the bottom, there's a "Manage permissions for your VM" section with links to understanding IAM roles and managing instance access.

Check the communication between these two instances belongs to two different networks using ping command. And it shows that no communication among these two instances.

The screenshot shows two terminal windows side-by-side. Both windows have the title "tsgcp\_user03@vpc2instance: ~ - Google Chrome".

**Left Terminal:**

```
tsgcp_user03@vpc2instance:~$ ssh cloud.google.com/projects/tarkshyas-1638768744041/zones/us-east1-b/instances/vpc2instance
Connected, host fingerprint: ssh-rsa 0 6f:31:7c:bf2:1e:92:51:bfa:20:4d:14:81:88:a0:f5:3c:82:97:51:1a:89:f0:50:26:b7:e7
Linux vpc2instance 4.19.0-18-cloud-amd64 #1 SMP Debian 4.19.208-1 (2021-06-26) x86_64

The programs included with the Debian GNU/Linux system are free software. The exact distribution terms for each program are described in the individual files in /usr/share/doc/*copyright.
```

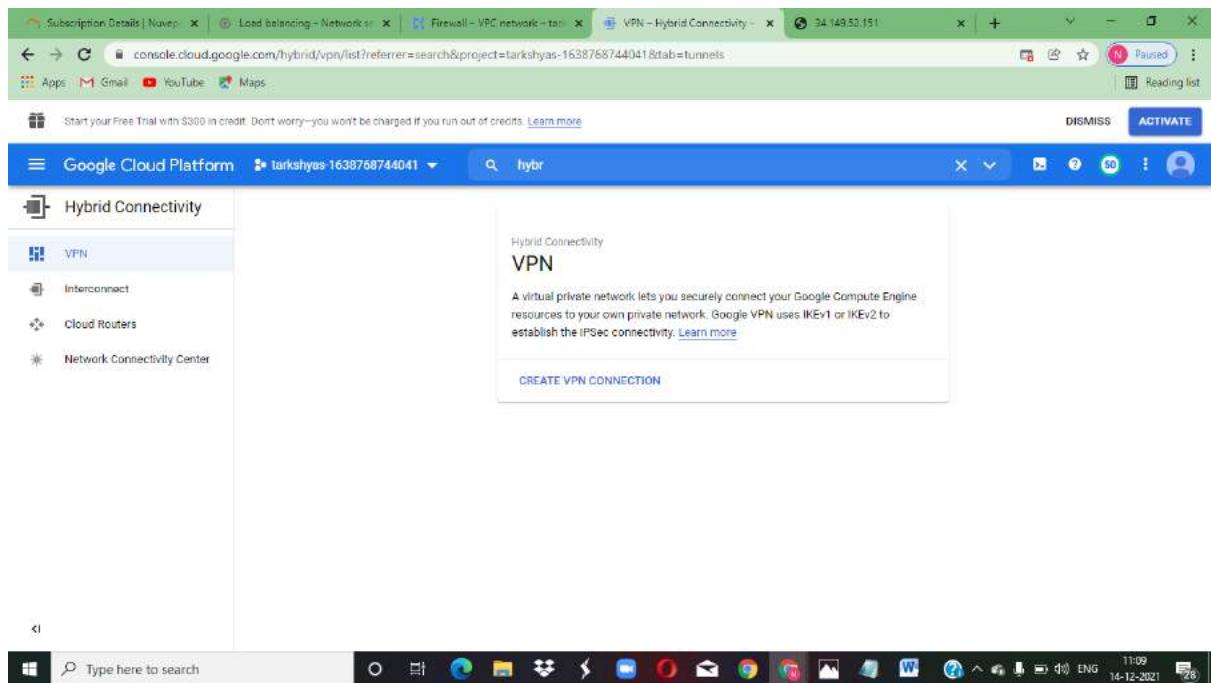
**Right Terminal:**

```
tsgcp_user03@vpc2instance:~$ ping 10.128.0.2
PING 10.128.0.2 (10.128.0.2) 56(84) bytes of data.

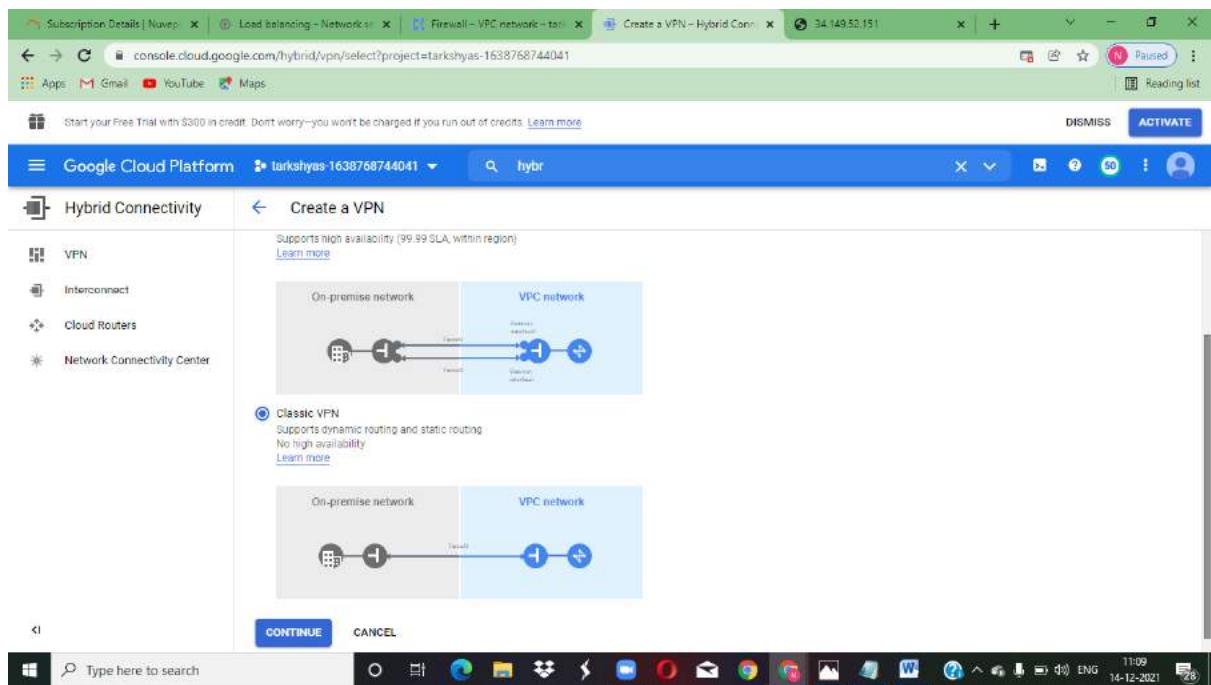
```

To make this communication possible we uses the VPN Networks. To create VPN Network click hybrid connectivity -> create VPN connection.

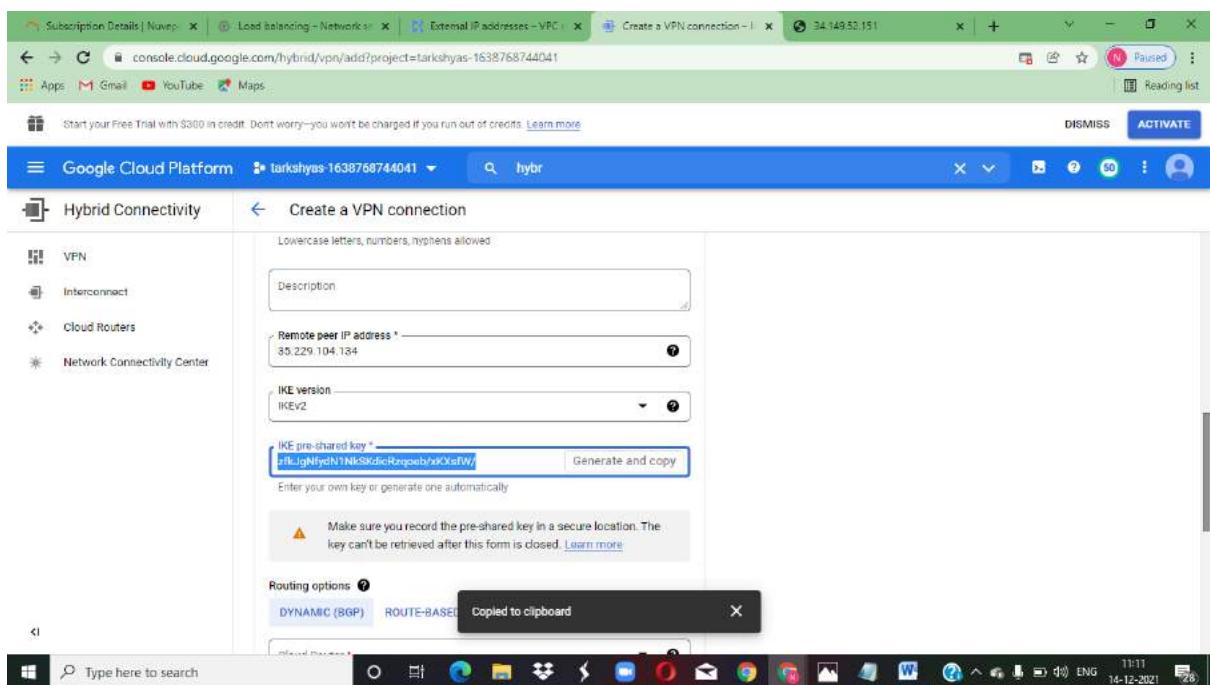
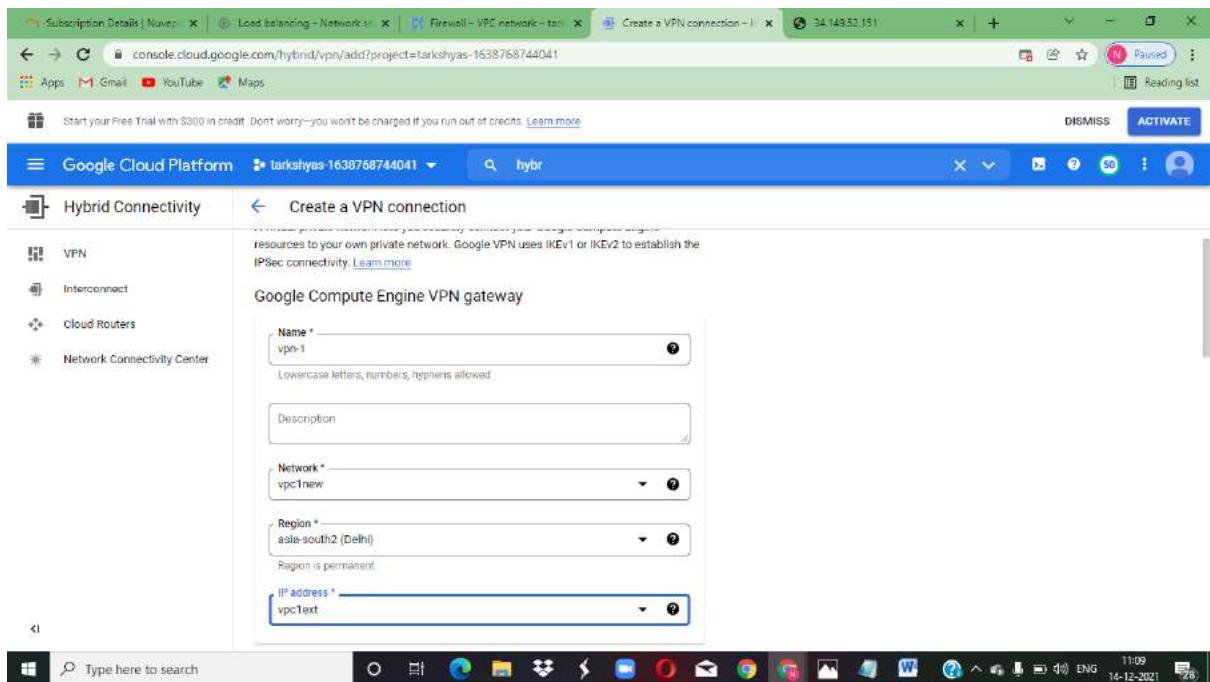
## VPN using Static Routing:



Click classic VPN.



On the create a VPN Connection page, specify name, description, network, region, and IP address.



Also configure Tunnel by specifying Name, Description, Remote peer ip address, IKE version  
And IKE Pre-shared Key.

The screenshot shows the Google Cloud Platform interface for creating a VPN connection. The left sidebar has 'Hybrid Connectivity' selected under 'VPN'. The main area is titled 'Create a VPN connection' with a back arrow. It shows fields for 'Region' (set to 'asia-south2 (Delhi)') and 'IP address' (set to 'vpc1text'). Below this is a 'Tunnels' section with a dropdown menu showing 'vpn-1-tunnel-1 (Remote peer IP address: 35.229.104.124)'. At the bottom are 'CREATE' and 'CANCEL' buttons, and an 'EQUIVALENT COMMAND LINE' section.

Thus first VPN is configured.

The screenshot shows the Google Cloud Platform interface for managing VPN gateways. The left sidebar has 'Hybrid Connectivity' selected under 'VPN'. The main area has tabs for 'CLOUD VPN TUNNELS', 'CLOUD VPN GATEWAYS' (which is selected), and 'PEER VPN GATEWAYS'. A 'CREATE VPN GATEWAY' button is visible. On the right, there's a 'Select a VPN gateway' panel with a message 'No VPN gateways selected.' Below it, a note says 'Labels help organize your resources (e.g., cost\_center\_sales or env/prod).'

Likewise create another VPN .

Subscription Details | Nuvep | Load balancing - Networks | VPC networks - VPC network | Create a VPN - Hybrid Conn | 34.149.52.151

console.cloud.google.com/hybrid/vpn/select?project=tarkshyas-1638768744041

DISMISS ACTIVATE

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

Google Cloud Platform tarkshyas-1638768744041 hybr

Hybrid Connectivity Create a VPN

VPN

On-premise network VPC network

Classic VPN

Supports dynamic routing and static routing  
No high availability

CONTINUE CANCEL

Type here to search 11:16 ENG 14-12-2021

Subscription Details | Nuvep | Load balancing - Networks | VPC networks - VPC network | Create a VPN connection - I | 34.149.52.151

console.cloud.google.com/hybrid/vpn/add?project=tarkshyas-1638768744041

DISMISS ACTIVATE

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

Google Cloud Platform tarkshyas-1638768744041 hybr

Hybrid Connectivity Create a VPN connection

VPN

A virtual private network lets you securely connect your Google Compute Engine resources to your own private network. Google VPN uses IKEv1 or IKEv2 to establish the IPsec connectivity. [Learn more](#)

Google Compute Engine VPN gateway

Name \* vpn-2

Description

Network \* vpc2new

Region \* us-east1 (South Carolina)

IP address \* vpc2ext

Type here to search 11:16 ENG 14-12-2021

The screenshot shows the Google Cloud Platform interface for creating a VPN connection. The left sidebar lists 'VPN', 'Interconnect', 'Cloud Routers', and 'Network Connectivity Center'. The main area is titled 'Create a VPN connection' and shows a form for a new item. The 'Name' field contains 'vpn-2-tunnel-1'. The 'Description' field is empty. The 'Remote peer IP address' field contains '34.131.18.32'. The 'IKE version' dropdown is set to 'IKEV2'. The 'IKE pre-shared key' field contains 'zfkjgNfydN1Nk8KdicRzqeb/xKXsfW/' and has a 'Generate and copy' button. A note below says 'Enter your own key or generate one automatically'. The status bar at the bottom shows '11:17 14-12-2021'.

The screenshot shows the configuration of the previously created VPN connection 'vpn-2-tunnel-1'. The 'Region' dropdown is set to 'us-east1 (South Carolina)'. The 'IP address' dropdown is set to 'vpc2ext'. In the 'Tunnels' section, there is a list with one item: 'vpn-2-tunnel-1 (Remote peer IP address: 34.131.18.32)'. Below this is a 'CREATE' button and a 'CANCEL' button. An 'EQUIVALENT COMMAND LINE' dropdown is also present. The status bar at the bottom shows '11:18 14-12-2021'.

Now 2 VPN Networks are created.

The screenshot shows the Google Cloud Platform Hybrid Connectivity interface. On the left sidebar, 'VPN' is selected. Under 'CLOUD VPN GATEWAYS', there are two entries:

- vpn-1**: IP address 34.131.18.32, VPC network vpc1new, Region asia-south2, VPN tunnels vpn-1-tunnel-1.
- vpn-2**: IP address 35.229.104.134, VPC network vpc2new, Region us-east1, VPN tunnels vpn-2-tunnel-1.

A right-hand panel titled 'Select a VPN gateway' displays a message: 'No VPN gateways selected.'

Now the communication between these two VM instances belongs to 2 different networks are possible.

The screenshot shows a terminal window with the title 'ssh\_user0@vpc2instance ~ - Google Chrome'. The command run is 'ssh user0@vpc2instance'. The terminal output shows a continuous stream of ICMP echo requests and responses between two hosts:

```

64 bytes from 10.2.0.2: icmp_seq=832 ttl=62 time=241 ms
64 bytes from 10.2.0.2: icmp_seq=833 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=834 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=835 ttl=62 time=241 ms
64 bytes from 10.2.0.2: icmp_seq=836 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=837 ttl=62 time=241 ms
64 bytes from 10.2.0.2: icmp_seq=838 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=839 ttl=62 time=241 ms
64 bytes from 10.2.0.2: icmp_seq=840 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=841 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=842 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=843 ttl=62 time=241 ms
64 bytes from 10.2.0.2: icmp_seq=844 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=845 ttl=62 time=241 ms
64 bytes from 10.2.0.2: icmp_seq=846 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=847 ttl=62 time=241 ms
64 bytes from 10.2.0.2: icmp_seq=848 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=849 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=850 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=851 ttl=62 time=241 ms
64 bytes from 10.2.0.2: icmp_seq=852 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=853 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=854 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=855 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=856 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=857 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=858 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=859 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=860 ttl=62 time=241 ms
64 bytes from 10.2.0.2: icmp_seq=861 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=862 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=863 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=864 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=865 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=866 ttl=62 time=241 ms
64 bytes from 10.2.0.2: icmp_seq=867 ttl=62 time=240 ms
64 bytes from 10.2.0.2: icmp_seq=868 ttl=62 time=240 ms

```

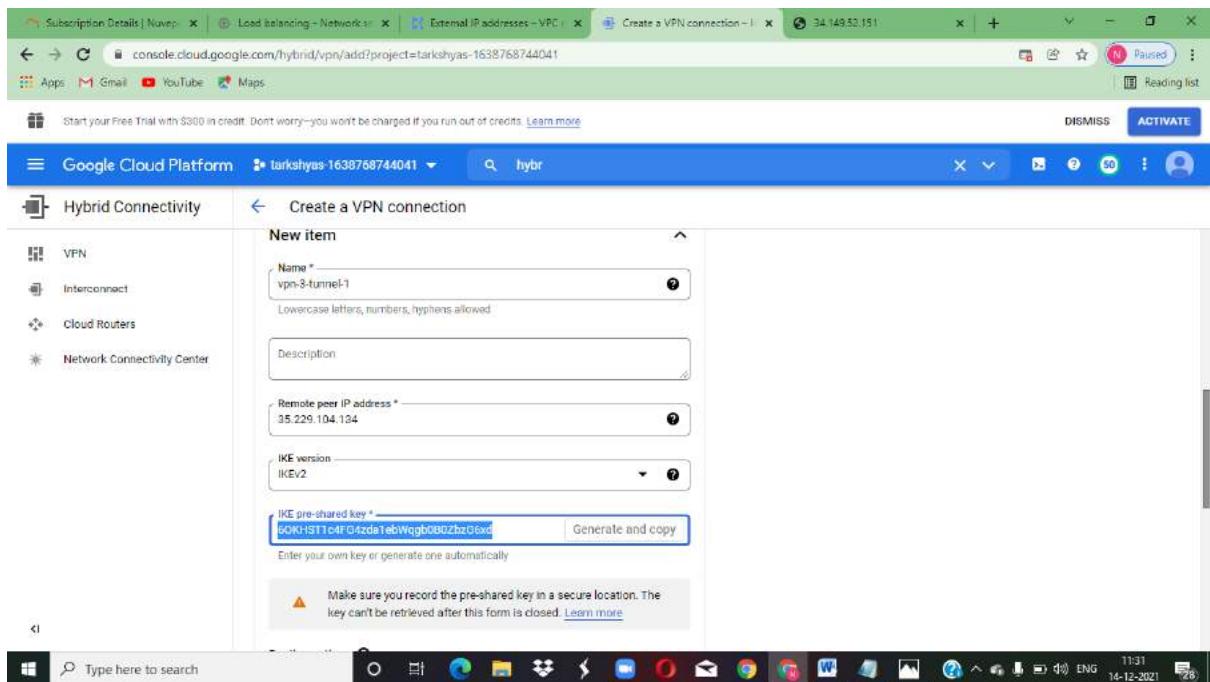
## VPN using Dynamic Routing

Click Classic VPN.

The screenshot shows the 'Create a VPN' wizard in the Google Cloud Platform. The left sidebar lists 'VPN', 'Interconnect', 'Cloud Routers', and 'Network Connectivity Center'. The main area shows two network connection diagrams: 'Classic VPN' and 'IPsec'. The 'Classic VPN' diagram shows an 'On-premise network' connected via 'Tunnel' to a 'VPC network'. The 'IPsec' diagram shows an 'On-premise network' connected via 'Virtual' to a 'VPC network'. Below the diagrams are 'CONTINUE' and 'CANCEL' buttons.

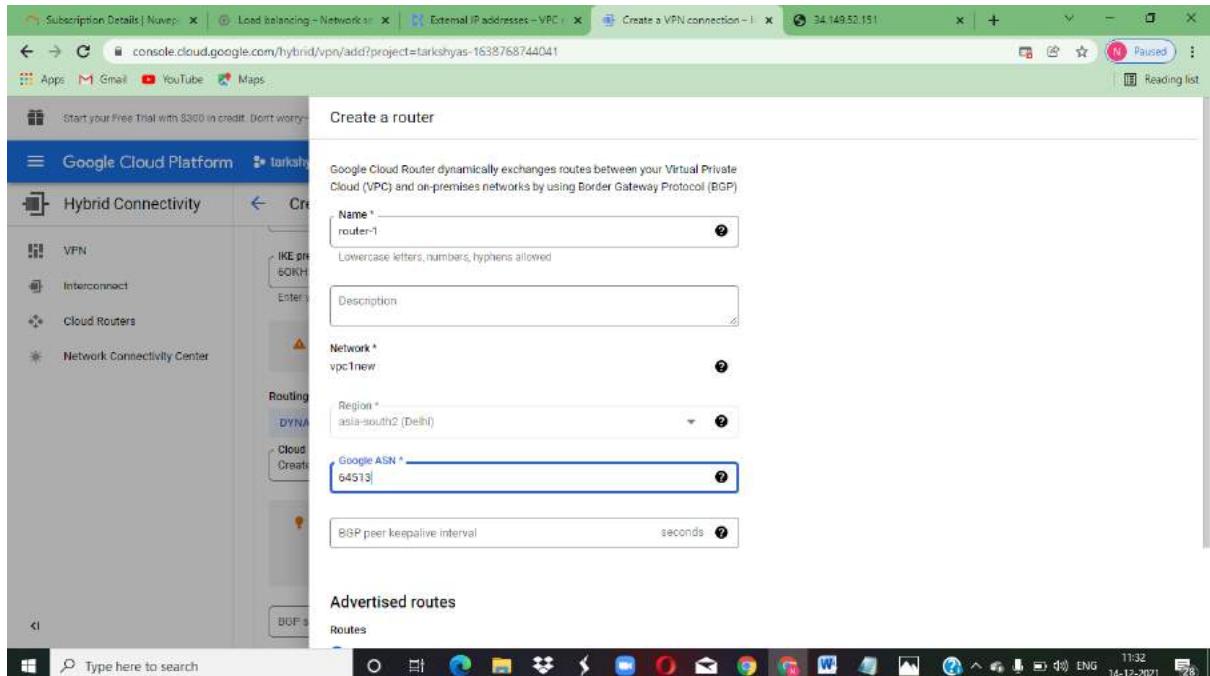
The screenshot shows the 'Create a VPN connection' wizard in the Google Cloud Platform. The left sidebar lists 'VPN', 'Interconnect', 'Cloud Routers', and 'Network Connectivity Center'. The main area is titled 'Google Compute Engine VPN gateway' and contains fields for 'Name' (set to 'vpn1-dynamic'), 'Description', 'Network' (set to 'vpcNew'), 'Region' (set to 'asia-south2 (Delhi)'), and 'IP address' (set to 'vpcText').

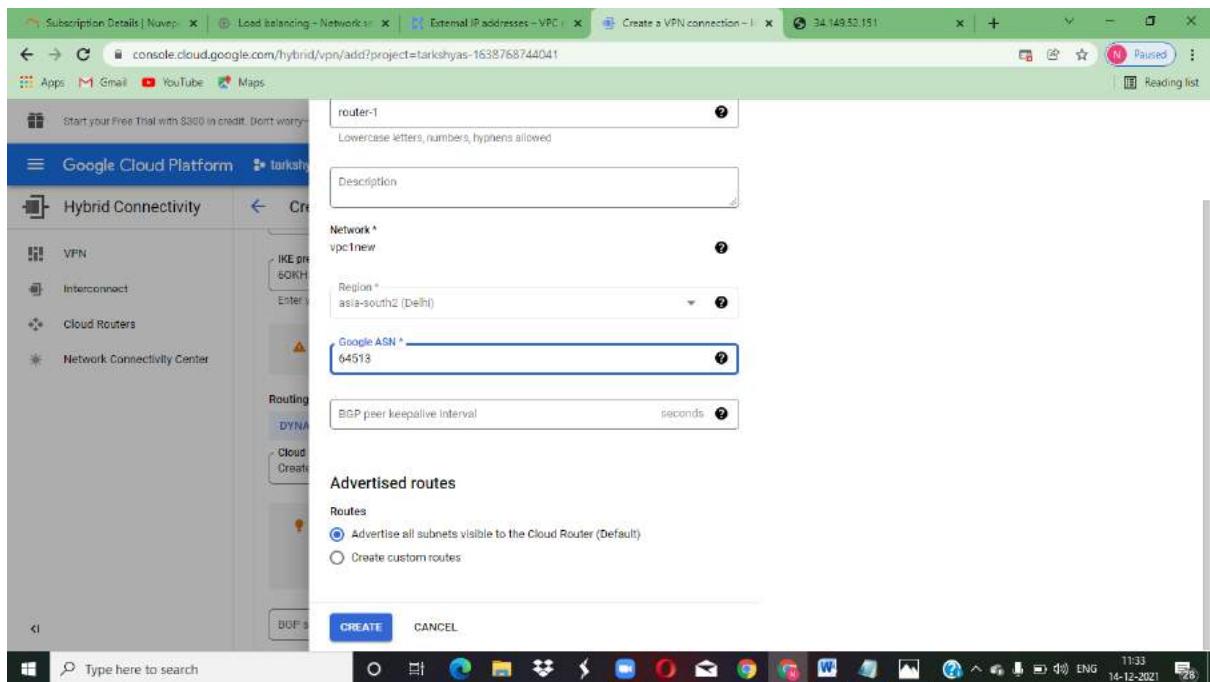
Specify Name, Description, Network, Region, IP address for gateway settings



Create new cloud router specify the following details:

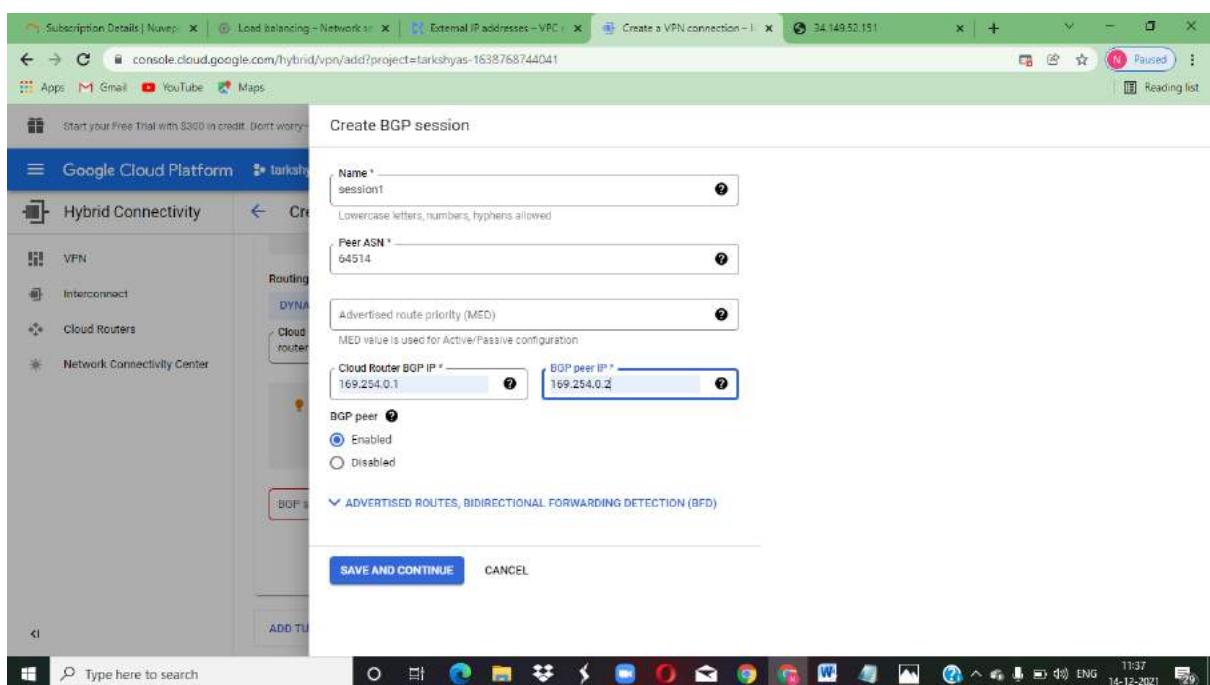
Name, Description, Google ASN (64512 through 65534, This is used for all BGP sessions managed by the cloud router). Click save and continue.

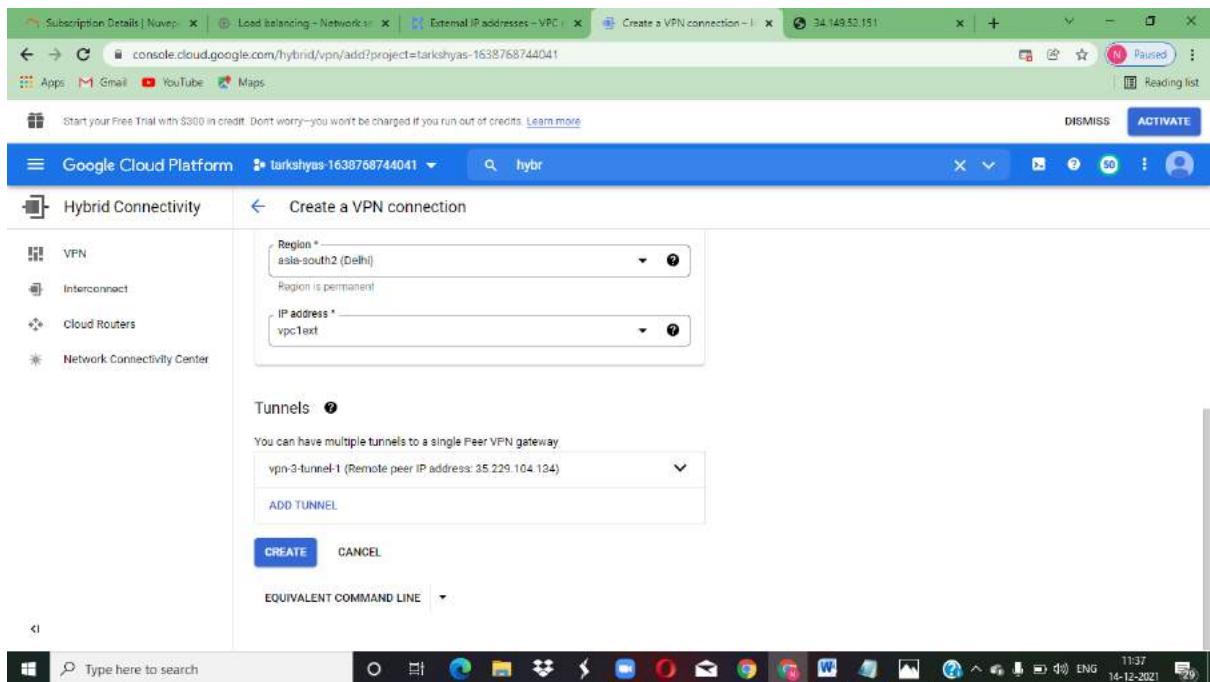




Create BGP Session.Specify Name, Peer ASN, Cloud router BGP IP and BGP peer IP.

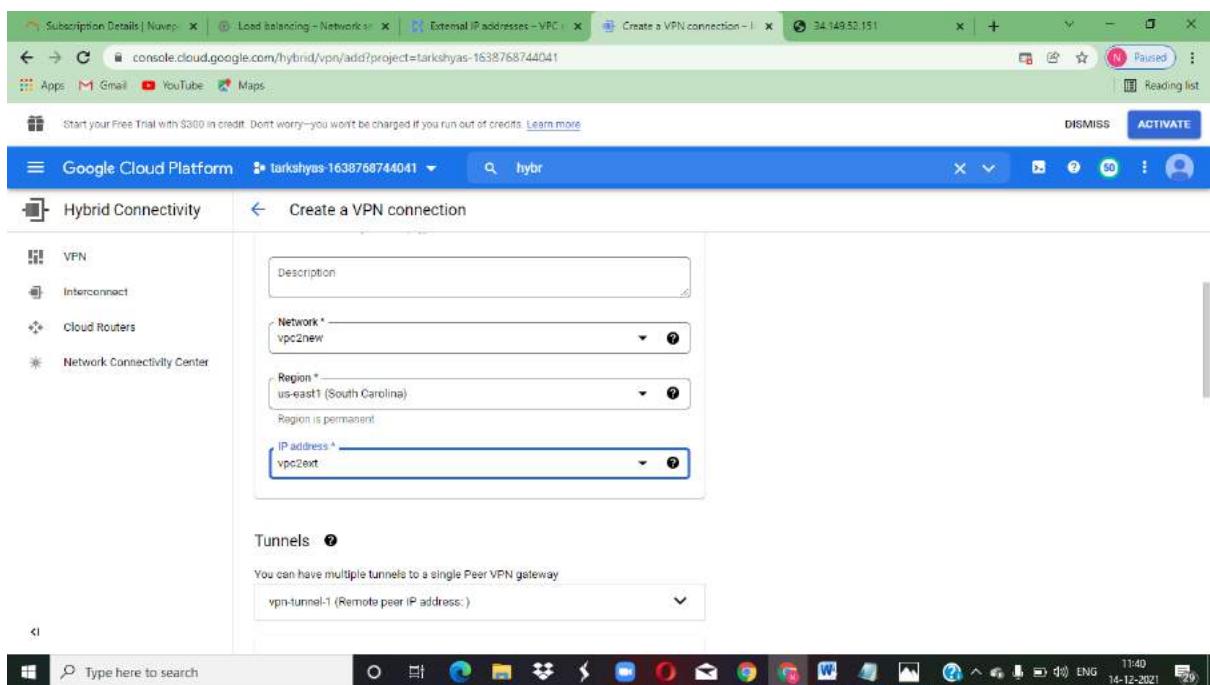
Click create.

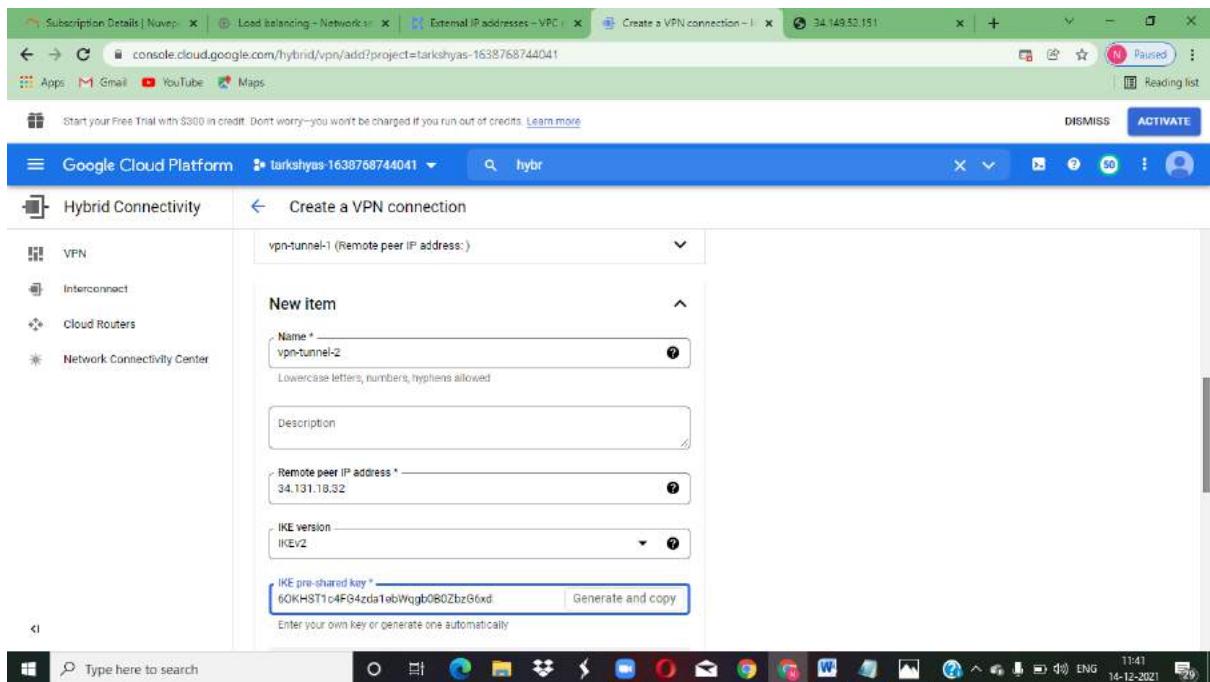




Create another VPN connection.

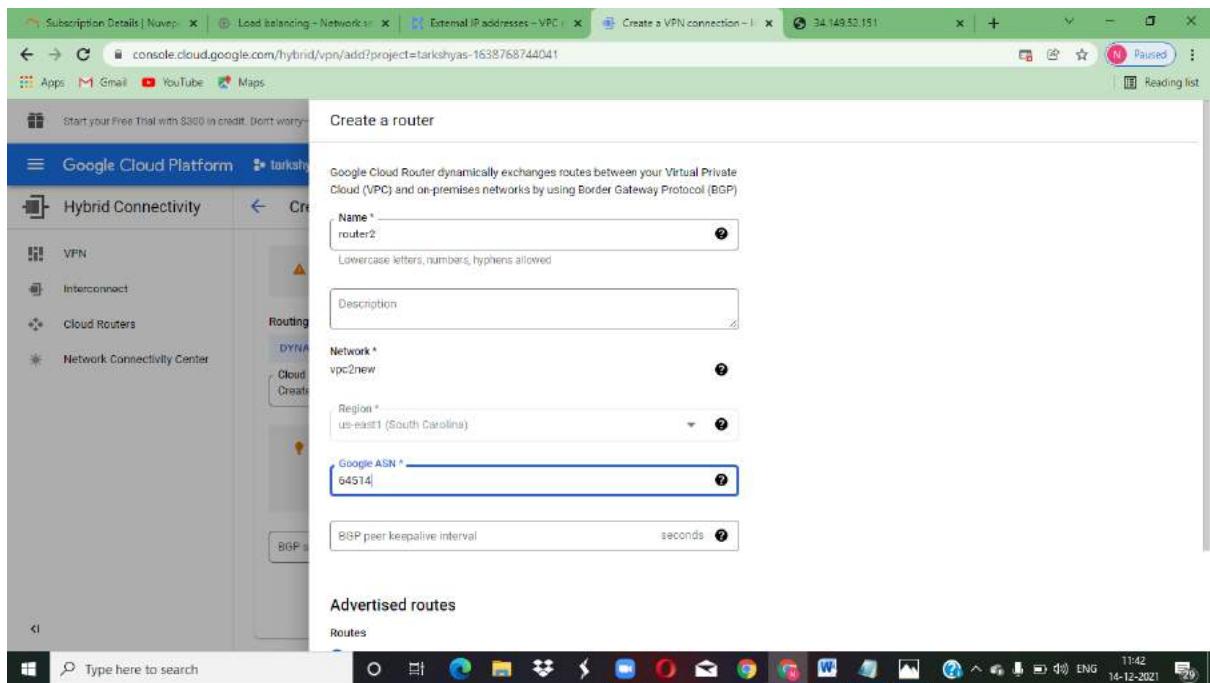
Specify Name, Description, Network, Region, IP address for gateway settings.





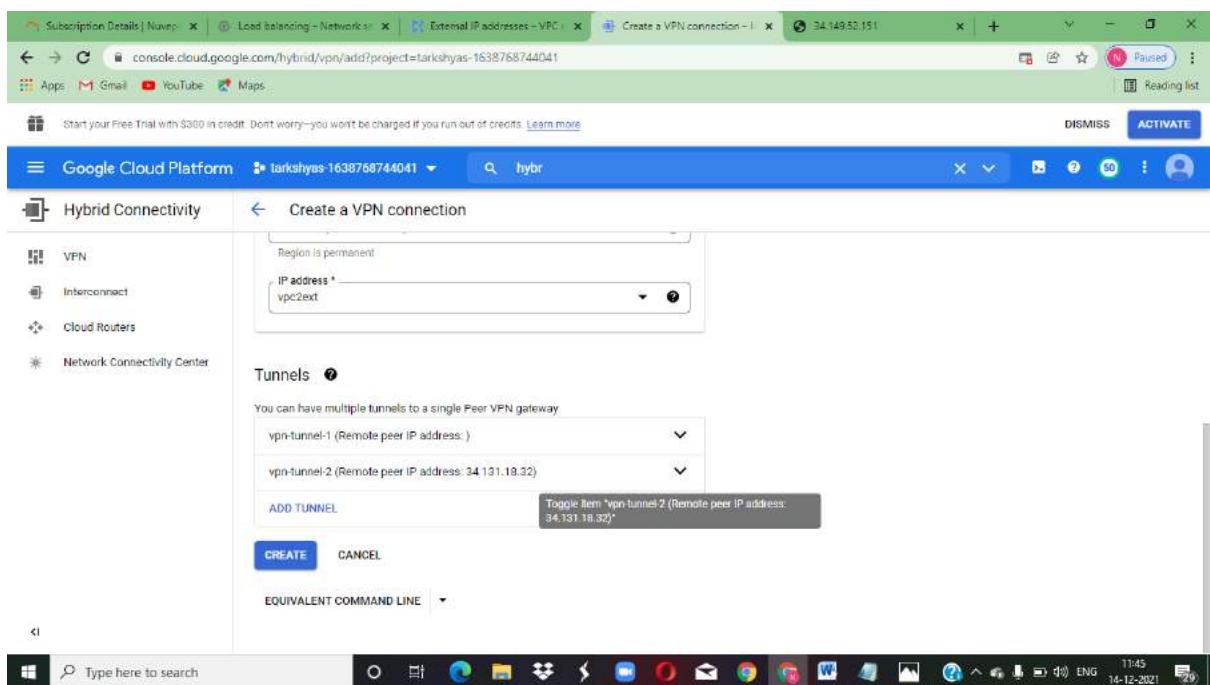
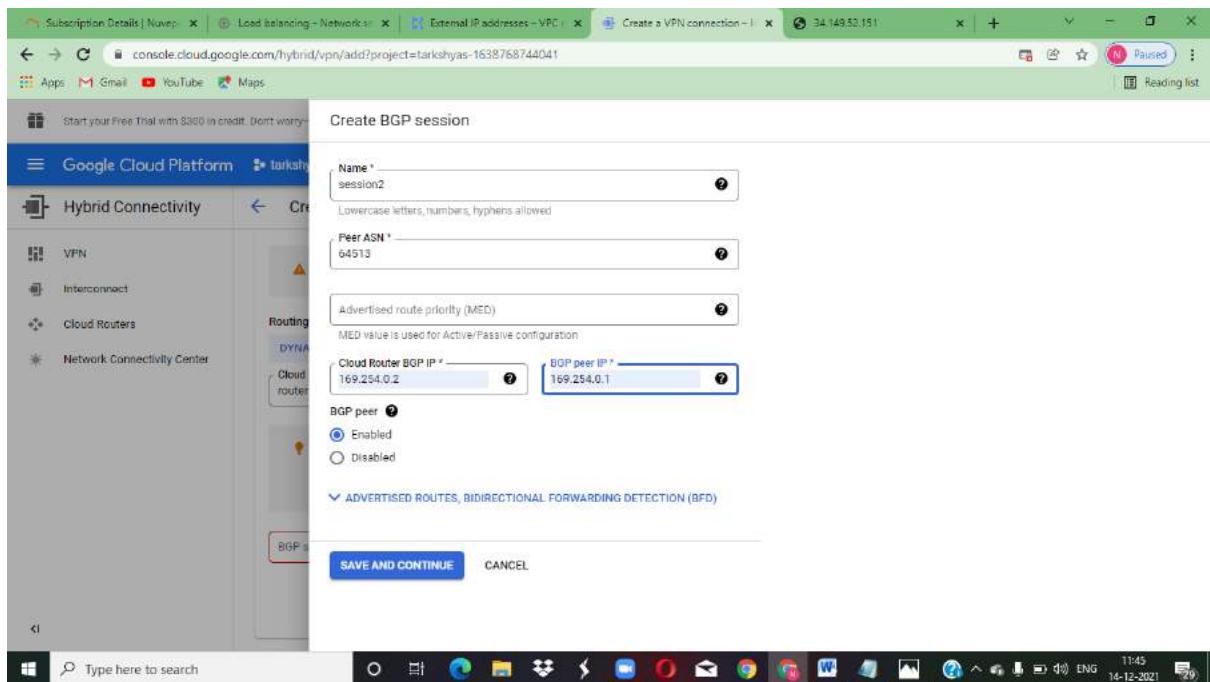
Create new cloud router specify the following details:

Name, Description, Google ASN (64512 through 65534, This is used for all BGP sessions managed by the cloud router). Click save and continue.



Create BGP Session.Specify Name, Peer ASN, Cloud router BGP IP and BGP peer IP.

Click create.



By using Dynamic routing we can connect two instances in two different networks .It has an advantage that can add and connect new systems and also communicate between them.

The image shows two terminal windows side-by-side. Both windows are titled 'tgcpc\_user03@vpc2instance ~ - Google Chrome' and show the command 'ping 10.128.0.2'. The left window's output is partially visible, while the right window's output is fully visible. The right window's output shows a series of ICMP echo requests (seq 1 to 31) being sent to the target IP at TTL=62, with round-trip times ranging from 241 ms to 260 ms.

```

ssh.cloud.google.com/projects/tarkshyas-1638768744041/zones/us-east1-b/instances/vpc2instance?aut...[REDACTED]
Connected, host fingerprint: sha-1:ea 0:60:31:7c:78:f2:[REDACTED] 19:81:68:9:fa:3e:88:82:9:73:61:1a:53:9:50:26:18:8:27
Linux vpc2instance 4.19.0-18-cloud-amd64 #1 SMP Debian 4.19.20
86_64

The programs included with the Debian GNU/Linux system are free
the exact distribution terms for each program are described in
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Dec 14 05:36:29 2021 from 35.235.240.18
tgcpc_user03@vpc2instance:~$ ping 10.128.0.2
PING 10.128.0.2 (10.128.0.2) 56(84) bytes of data.
64 bytes from 10.128.0.2: icmp_seq=1 ttl=62 time=260 ms
64 bytes from 10.128.0.2: icmp_seq=2 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=3 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=4 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=5 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=6 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=7 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=8 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=9 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=10 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=11 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=12 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=13 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=14 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=15 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=16 ttl=62 time=242 ms
64 bytes from 10.128.0.2: icmp_seq=17 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=18 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=19 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=20 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=21 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=22 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=23 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=24 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=25 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=26 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=27 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=28 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=29 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=30 ttl=62 time=241 ms
64 bytes from 10.128.0.2: icmp_seq=31 ttl=62 time=241 ms

```

### 3. Host a website given below using App engine standard environment.

<https://github.com/GoogleCloudPlatform/python-docs-samples/tree/main/appengine/standard/cloudsql>

Create App engine by click on Create application.

The image shows the Google Cloud Platform App Engine dashboard. The left sidebar lists various services like Services, Versions, Instances, Task queues, Cron jobs, Security scans, Firewall rules, Quotas, and Release Notes. The main area displays a 'Welcome to App Engine' message with a 'CREATE APPLICATION' button. On the right, there is a sidebar titled 'Recommended for you' with links to 'App Engine Quickstart', 'App Engine overview', 'Choosing an App Engine environment', 'Structuring web services in App Engine', 'Installing an SDK for App Engine', and 'App Engine pricing'.

Specify region as us-central.

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

DISMISS ACTIVATE

LEARN Home

Recommended for you

- App Engine Quickstart
- Deploy a sample application to App Engine using the `gcloud` command.
- App Engine overview
- Overview of the components of an application: services, versions, instances, application requests, and limits.
- Choosing an App Engine environment
- Run applications in App Engine using the standard or flexible environment.
- Structuring web services in App Engine
- Understand how to structure the services and related resources of your App Engine app.
- Installing an SDK for App Engine
- Set up your computer for developing, deploying, and managing your apps in App Engine.
- App Engine pricing
- Overview of the different pricing options for App Engine.

## Create MySQL instance .

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

DISMISS ACTIVATE

LEARN Home

Create a MySQL instance

Instance info

Instance ID \* temp

Use lowercase letters, numbers, and hyphens. Start with a letter.

Password \* \*\*\*\*

Get a password for the root user. [Learn more](#)

No password

Database version \* MySQL 8.0

Choose region and zonal availability

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)

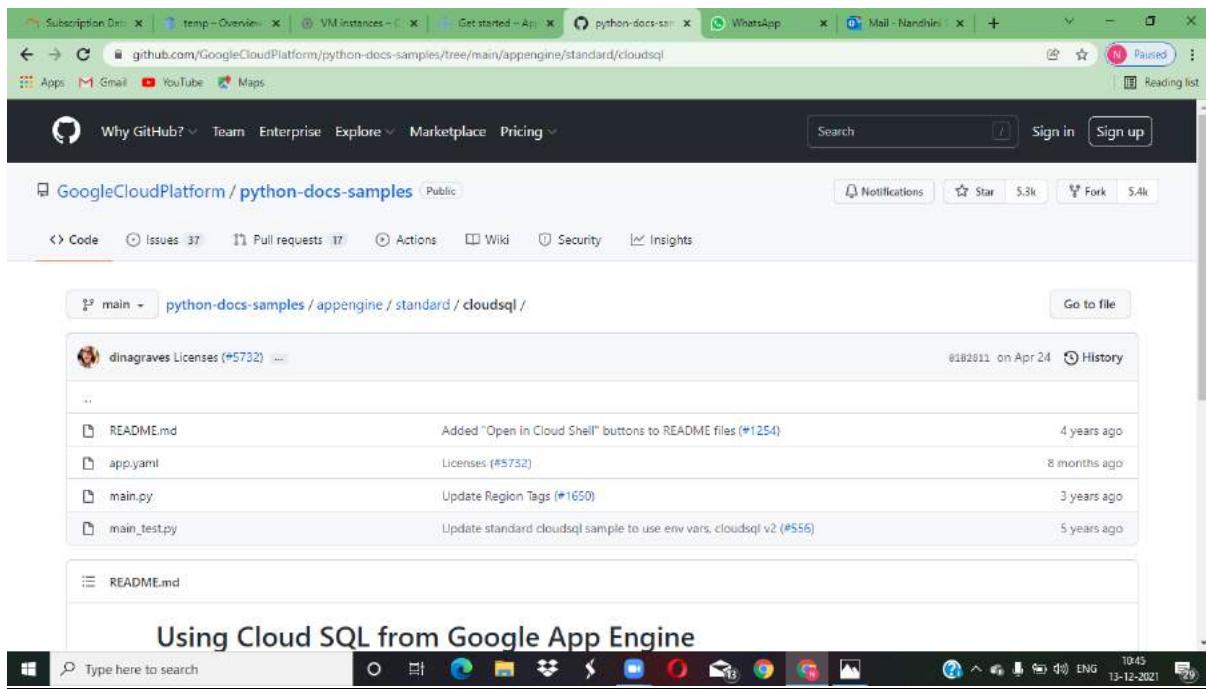
Summary

Region	us-central1 (Iowa)
DB Version	MySQL 8.0
vCPUs	4 vCPU
Memory	26 GB
Storage	100 GB
Network throughput (MB/s)	1,000 of 2,000
Disk throughput (MB/s)	Read: 48.0 of 240.0 Write: 48.0 of 240.0
IOPS	Read: 3,000 of 15,000 Write: 3,000 of 15,000
Connections	Public IP
Backup	Automated
Availability	Multiple zones (Highly available)
Point-in-time recovery	Enabled

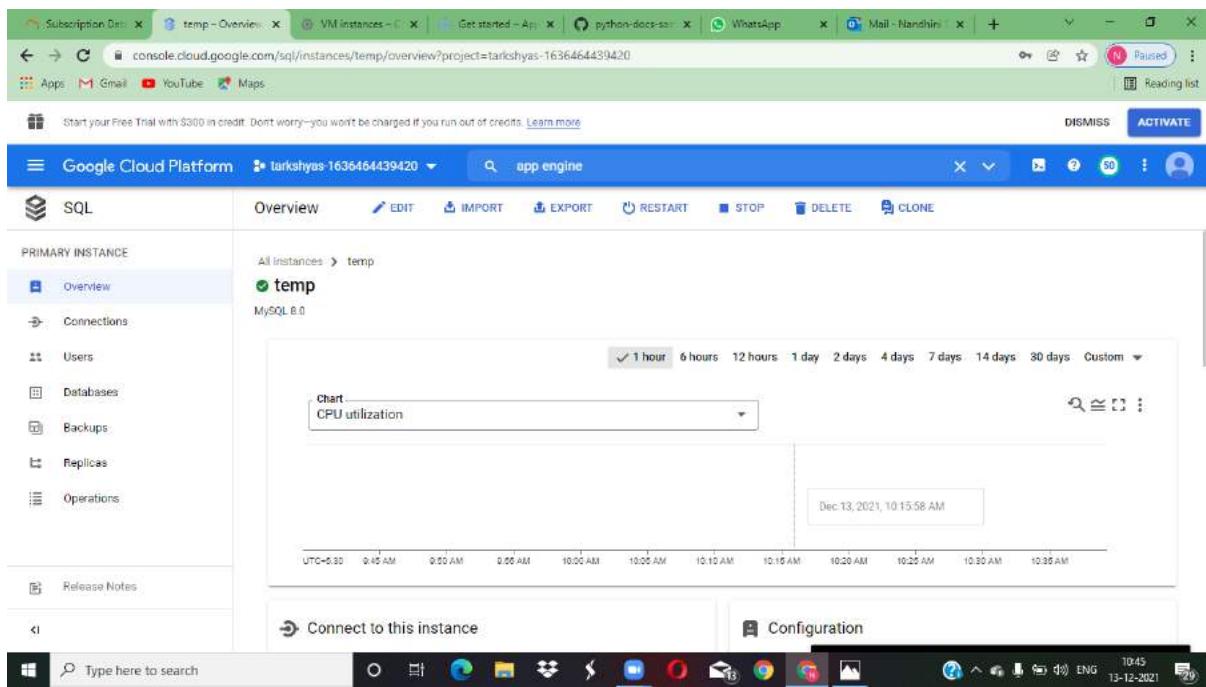
The screenshot shows a Google Cloud Platform interface for creating a MySQL instance. The top navigation bar includes tabs for Subscription Details, Create a MySQL instance, VM instances, WhatsApp, Mail, and a paused notification. Below the navigation is a banner about a free trial with \$300 credit. The main content area has a blue header "Google Cloud Platform" and "tarkshyas-1636464439420". A search bar and user profile are on the right. The main form is titled "Create a MySQL instance". It includes fields for Region (set to us-central1 (Iowa)), Zonal availability (Single zone selected), and Summary (MySQL 8.0 instance details like DB Version, vCPUs, Memory, Storage, Network throughput, Disk throughput, and IOPS). There's also a "Customize your instance" section and a "SHOW CONFIGURATION OPTIONS" dropdown. At the bottom are "CREATE INSTANCE" and "CANCEL" buttons.

Use the below github repository for hosting the application.

The screenshot shows a GitHub repository page for "GoogleCloudPlatform / python-docs-samples". The top navigation bar includes tabs for Subscription Data, temp - Overview, VM instances, Get started - App Engine, python-docs-samples, WhatsApp, Mail, and a paused notification. The main content area shows the repository's public status, code, issues (37), pull requests (17), actions, wiki, security, and insights. The code view shows a file named "main" with a list of commits. One commit is highlighted: "renovate-bot chore(deps): update dependency sendgrid to v6.9.2 (#7181)" made 11 days ago. Other commits listed include updates to analytics, angular, app\_identity, appstats, background, blobstore, and cloudsql. The bottom navigation bar includes a search bar and system status indicators.



SQL instance is created.



Create a new user account named newuser.

The screenshot shows the Google Cloud Platform SQL interface. On the left, a sidebar for the 'temp' instance lists 'Overview', 'Connections', 'Users' (which is selected), 'Databases', 'Backups', 'Replicas', and 'Operations'. The main area displays the 'Users' table for the 'temp' instance, which currently contains one row: 'root' with 'Host name' '% (any host)' and 'Authentication' 'Built-in'. A 'Release Notes' section is also present.

The screenshot shows the 'Add user account to instance temp' dialog. It asks 'Choose how to authenticate' between 'Built-in authentication' (selected) and 'Cloud IAM'. Under 'Built-in authentication', fields for 'User name \*' (set to 'newuser') and 'Password (Optional)' (set to '\*\*\*\*\*') are shown. Under 'Host name', 'Allow any host (%)' is selected. A note states that users created with built-in authentication have the same privileges as the root user. At the bottom are 'ADD' and 'CANCEL' buttons.

The screenshot shows the Google Cloud Platform SQL interface for a MySQL instance named 'temp'. The left sidebar has 'Users' selected under the 'PRIMARY INSTANCE' section. The main panel displays a table of users with two entries: 'newuser' and 'root'. Both users have 'Host name' set to '% (any host)' and 'Authentication' set to 'Built-in'. A 'User name' column header is shown at the top of the table.

User name	Host name	Authentication
newuser	% (any host)	Built-in
root	% (any host)	Built-in

To create new database click on databases.

The screenshot shows the Google Cloud Platform SQL interface for a MySQL instance named 'temp'. The left sidebar has 'Databases' selected under the 'PRIMARY INSTANCE' section. The main panel displays a table of databases with four entries: 'information\_schema', 'mysql', 'performance\_schema', and 'sys'. All databases are of type 'System' and use 'utf8\_general\_ci' collation and 'utf8' character set. A 'Name' column header is shown at the top of the table.

Name	Collation	Character set	Type
information_schema	utf8_general_ci	utf8	System
mysql	utf8_general_ci	utf8	System
performance_schema	utf8mb4_0900_ai_ci	utf8mb4	System
sys	utf8mb4_0900_ai_ci	utf8mb4	System

The screenshot shows the Google Cloud Platform SQL Databases interface. On the left, a sidebar lists options like Overview, Connections, Users, Databases (which is selected), Backups, Replicas, and Operations. The main area shows a MySQL 8.0 instance named 'temp'. A 'CREATE DATABASE' button is visible. To the right, a 'Create a database' form is open, showing the database name 'sampledb', character set 'utf8', and collation 'Default collation'. Below the form is a progress bar indicating 'CREATING A DATABASE...'.

New database temp is created.

The screenshot shows the same Google Cloud Platform SQL Databases interface after the database creation. The 'temp' database is now listed in the main table along with other system databases like information\_schema, mysql, performance\_schema, and sys. A notification at the bottom right of the screen says 'Created temp'.

Open Cloud shell and connect the new user.

Welcome to Cloud Shell! Type "help" to get started.  
Your Cloud Platform project in this session is set to tarkshyas-1636464439420.  
Use "gcloud config set project [PROJECT\_ID]" to change to a different project.  
tsgcp\_user03@cloudshell:~ (tarkshyas-1636464439420)\$ gcloud sql connect temp --user=newuser

DISMISS ACTIVATE

Search products and resources

CLOUD SHELL Terminal (tarkshyas-1636464439420) + Open Editor

Authorize Cloud Shell

gcloud is requesting your credentials to make a GCP API call  
Click to authorize this and future calls that require your credentials.

REJECT AUTHORIZE

Uploads and tarkshyas-1636464439420 operations

Created temp 10:45:03 AM GMT+5

Type here to search

Welcome to Cloud Shell! Type "help" to get started.  
Your Cloud Platform project in this session is set to tarkshyas-1636464439420.  
Use "gcloud config set project [PROJECT\_ID]" to change to a different project.  
tsgcp\_user03@cloudshell:~ (tarkshyas-1636464439420)\$ gcloud sql connect temp --user=newuser  
API [sqladmin.googleapis.com] not enabled on project [24265042199]. Would you like to enable and retry (this will take a few minutes)?  
(y/N)? y

Enabling service [sqladmin.googleapis.com] on project [24265042199]...  
Operation "operations/acf.p2-24265042199-f60a98a4-0459-4710-82ca-8014f30a7aec" finished successfully.  
Allowlisting your IP for incoming connection for 5 minutes...working

DISMISS ACTIVATE

Search products and resources

CLOUD SHELL Terminal (tarkshyas-1636464439420) + Open Editor

Uploads and tarkshyas-1636464439420 operations

Created temp 10:45:03 AM GMT+5

Type here to search

```
Subscription Data | temp - Database | VM instances - C | Get started - App | python-docs-samples | (1) WhatsApp | Mail - Nandini | + | - | X
← → C console.cloud.google.com/sql/instances/temp/databases?project=tarkshyas-1636464439420&cloudshell=true
DISMISS ACTIVATE
Google Cloud Platform tarkshyas-1636464439420 Search products and resources
CLOUD SHELL Terminal (tarkshyas-1636464439420) + Open Editor
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sampledb |
| sys |
+-----+
5 rows in set (0.20 sec)

mysql> use sampledb;
Database changed
mysql> create table demo(demo_id int,demo_text varchar(50));
Query OK, 0 rows affected (0.23 sec)

mysql> insert into demo values(11,"HELLO WORLD");
Query OK, 1 row affected (0.20 sec)

mysql> EXIT;
```

To deploy the application on app engine, install the gcloud component that include app engine extension for Python.

```
Subscription Data | temp - Database | VM instances - C | Get started - App | python-docs-samples | (1) WhatsApp | Mail - Nandini | + | - | X
← → C console.cloud.google.com/sql/instances/temp/databases?project=tarkshyas-1636464439420&cloudshell=true
DISMISS ACTIVATE
Google Cloud Platform tarkshyas-1636464439420 Search products and resources
CLOUD SHELL Terminal (tarkshyas-1636464439420) + Open Editor
Query OK, 1 row affected (0.20 sec)

mysql> EXIT;
Bye
tsgcp_user03@cloudshell:~ (tarkshyas-1636464439420)$ gcloud components install app-engine-python
ERROR: (gcloud.components.install)
You cannot perform this action because the Cloud SDK component manager
is disabled for this installation. You can run the following command
to achieve the same result for this installation:

sudo apt-get install google-cloud-sdk-app-engine-python

tsgcp_user03@cloudshell:~ (tarkshyas-1636464439420)$ ^C
tsgcp_user03@cloudshell:~ (tarkshyas-1636464439420)$ sudo apt-get install google-cloud-sdk-app-engine-python
*****
You are running apt-get inside of Cloud Shell. Note that your Cloud Shell
machine is ephemeral and no system-wide change will persist beyond session end.

To suppress this warning, create an empty ~/.cloudshell/no-apt-get-warning file.
The command will automatically proceed in 5 seconds or on any key.

Visit https://cloud.google.com/shell/help for more information.
*****
```

Clone the repository to your local machine and change to the directory that contains the sample code. Make changes on the code and deploy the app as follows:

```
threadsafe: yes

handlers:
- url: /
  script: main.app

libraries:
- name: MySQLdb
  version: "1.2.5"

# [START gae_python_mysql_env]
env_variables:
  CLOUDSQL_CONNECTION_NAME: tarkshyas-1636464439420:us-central1:temp
  CLOUDSQL_USER: newuser
  CLOUDSQL_PASSWORD: 1234
# [END gae_python_mysql_env]

-- INSERT --
```

```
db = MySQLdb.connect(
    host="",
    user=CLOUDSQL_USER,
    passwd=CLOUDSQL_PASSWORD)

return db

class MainPage(webapp2.RequestHandler):
    def get(self):
        """Simple request handler that shows all of the MySQL variables."""
        self.response.headers['Content-Type'] = 'text/plain'

        db = connect_to_cloudsql()
        cursor = db.cursor()
        cursor.execute("select @@version from dual")

        for r in cursor.fetchall():
            self.response.write("%s\n" % r)

app = webapp2.WSGIApplication([
    ('/', MainPage),
], debug=True)

# [END gae_python_mysql_app]
-- INSERT --
```

The screenshot shows a Google Cloud Platform terminal session. The user is in a Cloud Shell environment, specifically on a temporary instance named tarkshyas-1636464439420. They are running the command `gcloud app deploy` to upload an application from the directory `/home/tsgcp_user03/python-docs-samples/appengine/standard/cloudsql`. The application's configuration file, `app.yaml`, specifies a target service account and a target URL (`https://tarkshyas-1636464439420.uc.r.appspot.com`). The deployment process begins with a progress bar showing the upload of 5 files to Google Cloud Storage, reaching 100% completion. A confirmation message indicates the deployment was successful.

```
tsgcp_user03@cloudshell:~/python-docs-samples/appengine/standard/cloudsql (tarkshyas-1636464439420)$ vi app.yaml
tsgcp_user03@cloudshell:~/python-docs-samples/appengine/standard/cloudsql (tarkshyas-1636464439420)$ gcloud app deploy
Services to deploy:
descriptor:      [/home/tsgcp_user03/python-docs-samples/appengine/standard/cloudsql/app.yaml]
source:          [/home/tsgcp_user03/python-docs-samples/appengine/standard/cloudsql]
target project:  [tarkshyas-1636464439420]
target service:   [default]
target version:  [20211213t053923]
target url:      [https://tarkshyas-1636464439420.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]...
Uploading 5 files to Google Cloud Storage
20%
40%
60%
80%
100%
100%
File upload done.
```

This screenshot continues the terminal session from the previous one. The user has completed the deployment process, which resulted in a new service named [default] being deployed to the URL `https://tarkshyas-1636464439420.uc.r.appspot.com`. The terminal provides instructions for streaming logs and viewing the application in a web browser. It also notes that no browser was detected and provides a link to view the application via a web browser.

```
target service:      [default]
target version:     [20211213t054324]
target url:         [https://tarkshyas-1636464439420.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]...
Uploading 0 files to Google Cloud Storage
100%
File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://tarkshyas-1636464439420.uc.r.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
tsgcp_user03@cloudshell:~/python-docs-samples/appengine/standard/cloudsql (tarkshyas-1636464439420)$ gcloud app browse
Did not detect your browser. Go to this link to view your app:
https://tarkshyas-1636464439420.uc.r.appspot.com
tsgcp_user03@cloudshell:~/python-docs-samples/appengine/standard/cloudsql (tarkshyas-1636464439420)$
```

Click on the link to host the application



#### 4. Host a website given below using the App Engine Flexible Environment.

[https://github.com/GoogleCloudPlatform/python-docs-samples/tree/main/appengine/flexible/django\\_cloudsql](https://github.com/GoogleCloudPlatform/python-docs-samples/tree/main/appengine/flexible/django_cloudsql)

```
Welcome to Cloud Shell! Type "help" to get started.  
Your Cloud Platform project in this session is set to tarkshyas-1638768744041.  
Use "gcloud config set project [PROJECT_ID]" to change to a different project.  
tsgcp_user03@cloudshell:~ (tarkshyas-1638768744041)$ gcloud components install app-engine-python  
WARNING: (gcloud.components.install)  
You cannot perform this action because the Cloud SDK component manager  
is disabled for this installation. You can run the following command  
to achieve the same result for this installation:  
  
sudo apt-get install google-cloud-sdk-app-engine-python  
  
tsgcp_user03@cloudshell:~ (tarkshyas-1638768744041)$ sudo apt-get install google-cloud-sdk-app-engine-python  
*****  
You are running apt-get inside of Cloud Shell. Note that your Cloud Shell  
machine is ephemeral and no system-wide change will persist beyond session end.  
  
To suppress this warning, create an empty ~/.cloudshell/no-apt-get-warning file.  
The command will automatically proceed in 5 seconds or on any key.  
Visit https://cloud.google.com/shell/help for more information.  
*****  
[...]  
Uploads and tarkshyas-1638768744041 operations  
Created sql 1:42:17 PM GMT+5  
1452 14-12-2021
```

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

DISMISS ACTIVATE

Google Cloud Platform tarkshyas-1638768744041 cloud sql

CLOUD SHELL Terminal (tarkshyas-1638768744041) x + v

```
sudo apt-get install google-cloud-sdk-app-engine-python

tsgcp user03@cloudshell:~ (tarkshyas-1638768744041)$ sudo apt-get install google-cloud-sdk-app-engine-python
*****
You are running apt-get inside of Cloud Shell. Note that your Cloud Shell
machine is ephemeral and no system-wide change will persist beyond session end.

To suppress this warning, create an empty ~/.cloudshell/no-apt-get-warning file.
The command will automatically proceed in 5 seconds or on any key.

Visit https://cloud.google.com/shell/help for more information.
*****
Reading package lists... Done
Building dependency tree
Reading state information... Done
google-cloud-sdk-app-engine-python is already the newest version (365.0.1-0).
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
tsgcp user03@cloudshell:~ (tarkshyas-1638768744041)$ git clone \https://github.com/GoogleCloudPlatform/python-docs-samples
fatal: destination path 'python-docs-samples' already exists and is not an empty directory.
tsgcp user03@cloudshell:~ (tarkshyas-1638768744041)$ cd python-docs-samples
tsgcp user03@cloudshell:~/python-docs-samples (tarkshyas-1638768744041)$ cd appengine/flexible/django_cloudsql
tsgcp user03@cloudshell:~/python-docs-samples/appengine/flexible/django_cloudsql (tarkshyas-1638768744041)$ ls
app.yaml manage.py mysite noxfile_config.py polls README.md requirements-test.txt requirements.txt
tsgcp user03@cloudshell:~/python-docs-samples/appengine/flexible/django_cloudsql (tarkshyas-1638768744041)$ vi app.yaml
```

Type here to search

14:55 14-12-2021

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

DISMISS ACTIVATE

Google Cloud Platform tarkshyas-1638768744041 cloud sql

CLOUD SHELL Terminal (tarkshyas-1638768744041) x + v

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to tarkshyas-1638768744041.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
tsgcp user03@cloudshell:~ (tarkshyas-1638768744041)$ gcloud components install app-engine-python
gcloud: (gcloud.components.install)
You cannot perform this action because the Cloud Shell
is disabled for this installation. You can run the command
to achieve the same result for this installation.

sudo apt-get install google-cloud-sdk-app-engine-python

tsgcp user03@cloudshell:~ (tarkshyas-1638768744041)$
*****
You are running apt-get inside of Cloud Shell. Note that your Cloud Shell
machine is ephemeral and no system-wide change will persist beyond session end.

To suppress this warning, create an empty ~/.cloudshell/no-apt-get-warning file.
The command will automatically proceed in 5 seconds or on any key.

tsgcp user03@cloudshell:~/python-docs-samples/appengine/flexible/django_cloudsql (tarkshyas-1638768744041)$ ls
app.yaml manage.py mysite noxfile_config.py polls README.md requirements-test.txt requirements.txt
tsgcp user03@cloudshell:~/python-docs-samples/appengine/flexible/django_cloudsql (tarkshyas-1638768744041)$ vi app.yaml
tsgcp user03@cloudshell:~/python-docs-samples/appengine/flexible/django_cloudsql (tarkshyas-1638768744041)$ gcloud app deploy
```

Authorize Cloud Shell

gcloud is requesting your credentials to make a GCP API call.

Click to authorize this and future calls that require your credentials.

REJECT AUTHORIZE

Type here to search

14:58 14-12-2021

You are running apt-get inside of Cloud Shell. Note that your Cloud Shell machine is ephemeral and no system-wide change will persist beyond session end.

To suppress this warning, create an empty `~/.cloudshell/no-apt-get-warning` file. The command will automatically proceed in 5 seconds or on any key.

```
tsgcp_user03@cloudshell:~/python-docs-samples/appengine/flexible/django_cloudsql (tarkshyas-1638768744041)$ ls app.yaml manage.py mysite noxfile_config.py polls README.md requirements-test.txt requirements.txt
tsgcp_user03@cloudshell:~/python-docs-samples/appengine/flexible/django_cloudsql (tarkshyas-1638768744041)$ vi app.yaml
tsgcp_user03@cloudshell:~/python-docs-samples/appengine/flexible/django_cloudsql (tarkshyas-1638768744041)$ gcloud app deploy
Services to deploy:
```

descriptor:	[/home/tsgcp_user03/python-docs-samples/appengine/flexible/django_cloudsql/app.yaml]
source:	[/home/tsgcp_user03/python-docs-samples/appengine/flexible/django_cloudsql]
target project:	[tarkshyas-1638768744041]
target service:	[default]
target version:	[20211214t092039]
target url:	[https://tarkshyas-1638768744041.uc.r.appspot.com]
target service account:	[App Engine default service account]

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

```
Enabling service [appenginetest.googleapis.com] on project [tarkshyas-1638768744041]...
```

14:58 14-12-2021

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

DISMISS ACTIVATE

Google Cloud Platform tarkshyas-1638768744041 cloud sql

CLOUD SHELL Terminal (tarkshyas-1638768744041) +

```
74bccef7f7402: Waiting
bc9e931c388e: Waiting
b49bce339f97: Waiting
20896f2c3ddd: Waiting
dcb7197db903: Waiting
007d7553d285: Layer already exists
16919ab9eca: Layer already exists
74bccef7f7402: Layer already exists
bc9e931c388e: Layer already exists
b03a959c0d4d: Pushed
161997b5117a: Pushed
7b60c69caf34: Layer already exists
20896f2c3ddd: Layer already exists
3bbec54fac0c: Layer already exists
844d958e8cbe: Layer already exists
4006ffa4c683: Layer already exists
84ff92691f90: Layer already exists
b49bce339f97: Layer already exists
dcb7197db903: Layer already exists
f7998300a6af: Pushed
f391dc518d1c: Pushed
latest: digest: sha256:d4a2f25e28f23a65bccdc07ebc6222f8d59e339899071660a7938cd1714a2786 size: 3670
DONE
```

Updating service [default] (this may take several minutes)...working

15:01 14-12-2021

```
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

START runtime
START gaeflex py_django_app.yaml
runtime: python
env: flex
entrypoint: gunicorn -b :$PORT mysite.wsgi

beta_settings:
  cloud_sql_instances: tarkshyas-1638768744041:us-central1:sql

runtime_config:
  python_version: 3
END gaeflex py_django_app.yaml
END runtime
app.yaml" 27L, 870C
```

```
tagcp_user03@cloudshell:~/python-docs-samples/appengine/flexible (tarkshyas-1638768744041)$ cd hello_world
tagcp_user03@cloudshell:~/python-docs-samples/appengine/flexible/hello_world (tarkshyas-1638768744041)$ ls
app.yaml main.py main-test.py requirements-test.txt requirements.txt
tagcp_user03@cloudshell:~/python-docs-samples/appengine/flexible/hello_world (tarkshyas-1638768744041)$ vi app.yaml
```

The screenshot shows a Google Cloud Platform Cloud Shell interface. The terminal window displays a Python script named `main.py` which contains a simple Flask application. The script includes a `hello` endpoint that returns a greeting message. A note at the bottom indicates that the app is designed to run locally or via a webserver like Gunicorn.

```
# Licensed under the Apache License, Version 2.0
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# [START gae_flex_quickstart]
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello():
    """Return a friendly HTTP greeting."""
    return "Hello, World!"

if __name__ == '__main__':
    # This is used when running locally only. When deploying to Google App
    # Engine, a webserver process such as Gunicorn will serve the app.
    main.py 32L, 1034C
```

The screenshot shows a Google Cloud Platform Cloud Shell interface. The terminal window displays a deployment command using the `gcloud app deploy` command. The command specifies the source directory as `/home/tsgcp_user03/python-docs-samples/appengine/flexible/hello_world`, the target project as `tarkshyas-1638768744041`, and the target service as `hello_world`. The deployment is successful, and the URL for the deployed application is provided.

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to tarkshyas-1638768744041.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
tsgcp_user03@cloudshell:~ (tarkshyas-1638768744041)$ cd python-docs-samples
tsgcp_user03@cloudshell:~/python-docs-samples/appengine/flexible/hello_world (tarkshyas-1638768744041)$ gcloud app deploy
tsgcp_user03@cloudshell:~/python-docs-samples/appengine/flexible/hello_world (tarkshyas-1638768744041)$ vi app.yaml
tsgcp_user03@cloudshell:~/python-docs-samples/appengine/flexible/hello_world (tarkshyas-1638768744041)$ vi main.py
tsgcp_user03@cloudshell:~/python-docs-samples/appengine/flexible/hello_world (tarkshyas-1638768744041)$ gcloud app deploy
Services to deploy:
descriptor:          [/home/tsgcp_user03/python-docs-samples/appengine/flexible/hello_world/app.yaml]
source:              [/home/tsgcp_user03/python-docs-samples/appengine/flexible/hello_world]
target project:      [tarkshyas-1638768744041]
target service:      [default]
target version:      [20211214t103426]
target url:          [https://tarkshyas-1638768744041.uc.r.appspot.com]
target service account: [App Engine default service account]

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

```
4006ffa4ac683: Layer already exists
844d958e8cbe: Layer already exists
84ff92691f90: Layer already exists
b49bce339f97: Layer already exists
dcb7197db903: Layer already exists
98e87b35a520: pushed
20e6622af9c1: pushed
2c5feabbd0025: Pushed
6ef349bd0de4: Pushed
latest: digest: sha256:9b8269fa7ea50482e5113259491daeeb78a6a52e97ddfe4df43382b753a70b3f size: 3669
DONE
-----
Updating service [default] (this may take several minutes)...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://tarkshyas-1638768744041.uc.r.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
tsgcp_user03@cloudshell:~/python-docs-samples/appengine/flexible/hello_world (tarkshyas-1638768744041)$
```

cloud\_sql\_proxy\_x86.exe Canceled Show all X

Type here to search 16:13 ENG 14-12-2021

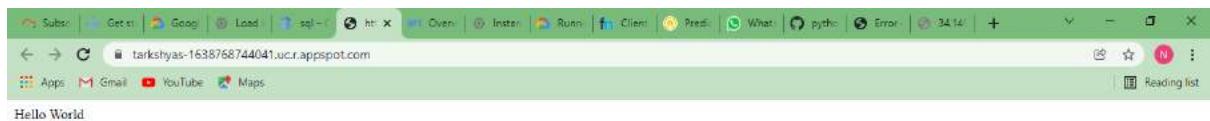
```
b49bce339f97: Layer already exists
dcb7197db903: Layer already exists
98e87b35a520: pushed
20e6622af9c1: Pushed
2c5feabbd0025: pushed
6ef349bd0de4: Pushed
latest: digest: sha256:9b8269fa7ea50482e5113259491daeeb78a6a52e97ddfe4df43382b753a70b3f size: 3669
DONE
-----
Updating service [default] (this may take several minutes)...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://tarkshyas-1638768744041.uc.r.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
tsgcp_user03@cloudshell:~/python-docs-samples/appengine/flexible/hello_world (tarkshyas-1638768744041)$ gcloud app browse
Did not detect your browser. Go to this link to view your app:
https://tarkshyas-1638768744041.uc.r.appspot.com
tsgcp_user03@cloudshell:~/python-docs-samples/appengine/flexible/hello_world (tarkshyas-1638768744041)$
```

cloud\_sql\_proxy\_x86.exe Canceled Show all X

Type here to search 16:14 ENG 14-12-2021



## 5. Create a cloud function to host a http function.

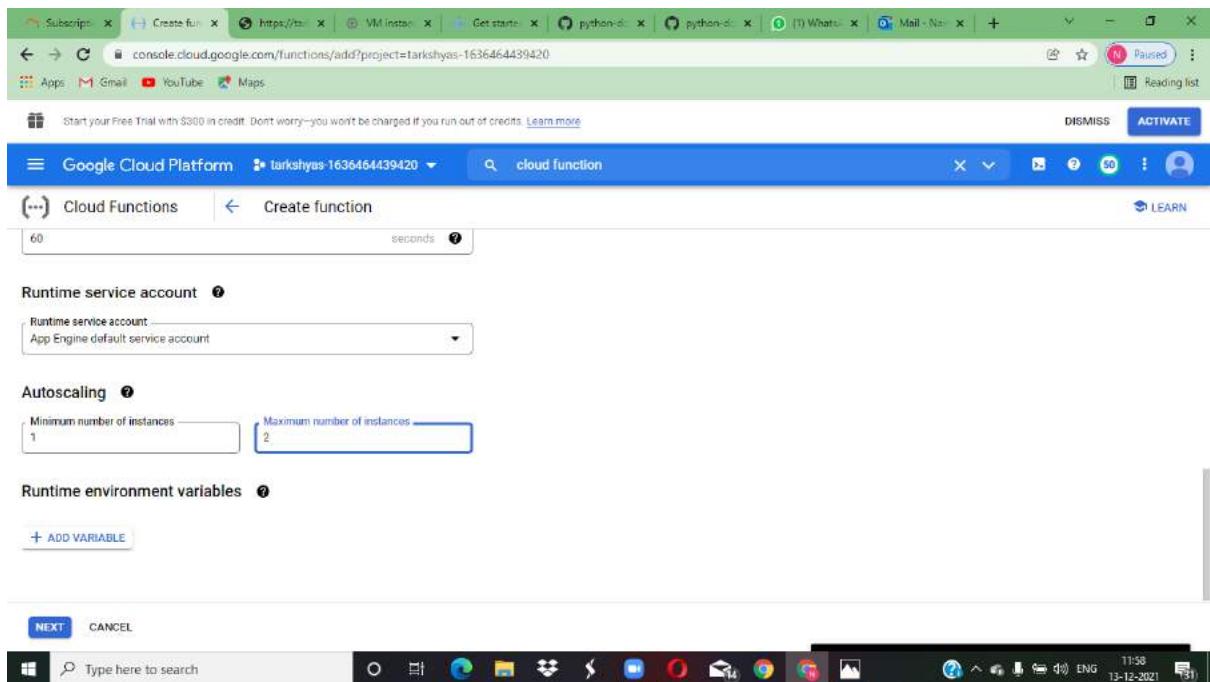
Click on create function

Specify function name, region and http trigger.

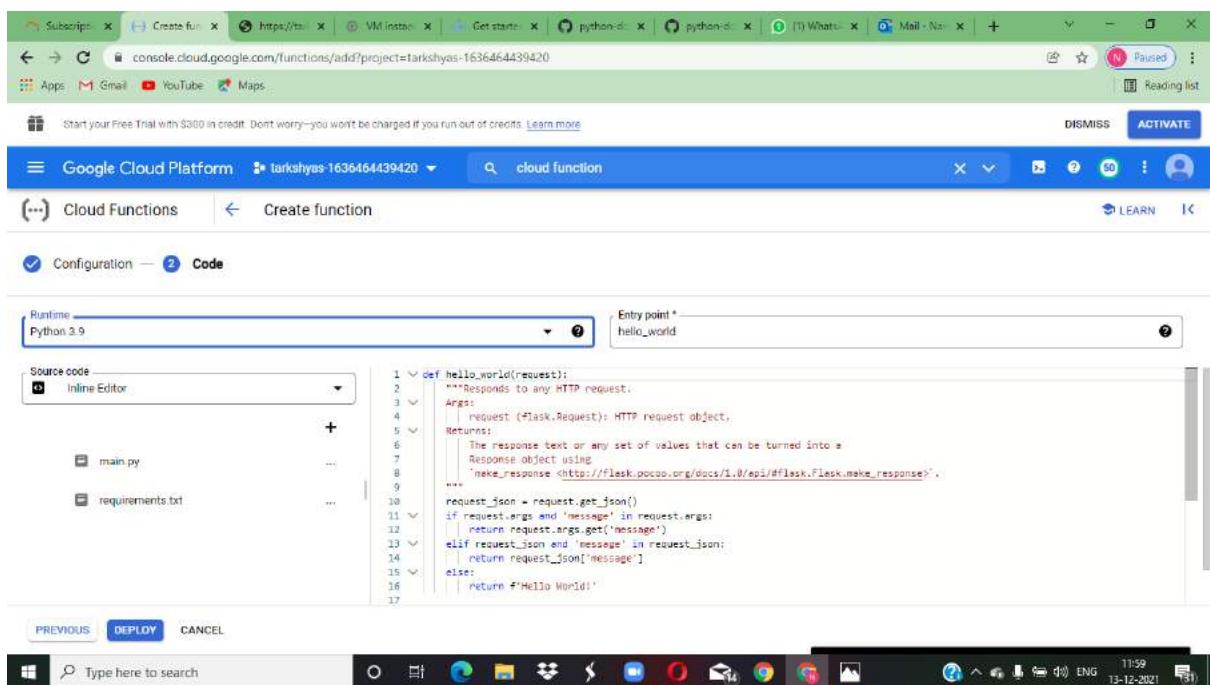
The screenshot shows the 'Create function' wizard in the Google Cloud Platform console. The 'Configuration' step is active. In the 'Basics' section, the 'Function name' field contains 'function1' and the 'Region' dropdown is set to 'us-central1'. In the 'Trigger' section, 'HTTP' is selected as the trigger type, and the 'Trigger type' dropdown also shows 'HTTP'. At the bottom, there are 'NEXT' and 'CANCEL' buttons.

The screenshot shows the 'Create function' wizard on the Google Cloud Platform. The current step is 'Trigger'. A sidebar on the left lists 'Cloud Functions' and 'Create function'. The main area shows an 'HTTP' trigger configuration with a 'Trigger URL' of <https://us-central1-tarkshyas-1636464439420.cloudfunctions.net/function1>. Below it, 'Authentication' settings allow unauthenticated invocations and require HTTPS. A blue 'EDIT' button is at the bottom. A dropdown menu for 'Runtime, build, connections and security settings' is open. At the bottom are 'NEXT' and 'CANCEL' buttons.

The screenshot shows the 'Create function' wizard on the Google Cloud Platform. The current step is 'Runtime service account'. It includes fields for 'Runtime service account' (set to 'App Engine default service account') and 'Autoscaling' (with 'Minimum number of instances' at 0 and 'Maximum number of instances' at 3000). A 'Runtime environment variables' section with a '+ ADD VARIABLE' button is also present. At the bottom are 'NEXT' and 'CANCEL' buttons.



Deploy the code written in Python.



Cloud function created.

The screenshot shows the Google Cloud Platform Cloud Functions list page. The URL in the browser is `console.cloud.google.com/functions/list?project=tarkshyas-1636464439420`. The page title is "Google Cloud Platform" and the search bar contains "cloud function". The main table lists one function:

Name	Region	Trigger	Runtime	Memory allocated	Executed function	Last deployed	Authentication	Actions
function-1	us-central1	HTTP	Python 3.9	256 MB	hello_world	Dec 13, 2021, 12:00:41 PM	Allow unauthenticated	⋮



Click on Trigger to deploy the http function.

The screenshot shows the Google Cloud Platform Cloud Function details page for "function-1". The URL is `console.cloud.google.com/functions/details/us-central1/function-1?project=tarkshyas-1636464439420&tab=trigger`. The page title is "Function details" and the search bar contains "cloud function". The main table shows the function details, and the "TRIGGER" tab is selected. Under the "HTTP" section, the "Trigger URL" is listed as `https://us-central1-tarkshyas-1636464439420.cloudfunctions.net/function-1`.





We can also perform testing on the code as follows:

A screenshot of the Google Cloud Platform Cloud Functions console. The URL in the browser is 'console.cloud.google.com/functions/details/us-central1/function-1?project=tarkshyas-1636464439420&tab=testing'. The page shows 'Function details' for 'function-1'. The 'TESTING' tab is selected. Under 'Triggering event', there is a code editor with the following content:

```
1  "message": "hi team"
2
```

Below the code editor is a blue button labeled '(-->) TEST THE FUNCTION'. A note below the button says 'Testing in the Cloud Console has a 5 minute timeout. Note that this is different from the limit set in the function configuration.' At the bottom, there is an 'Output' section showing 'Complete' and the message '\$ hi team'. The browser's status bar at the bottom indicates the time is 12:01 on December 13, 2021.

## 6. Host a website using Cloud run.

Click on Create Service.

Select the Container image.

The screenshot shows the Google Cloud Platform Cloud Run service creation interface. On the left, the main form has 'Service settings' selected. Under 'Deploy one revision from an existing container image', the 'Container Image URL' field contains 'us.gcr.io/tarkshyas-1636464439420/cad-site'. A 'SELECT' button is next to it. Below this, there's a 'TEST WITH A SAMPLE CONTAINER' section with a note about listening on port 8080. The 'Region' dropdown is set to 'us-central1 (Iowa)'. On the right, a modal dialog titled 'Select container image' is open, showing a list of container images under 'CONTAINER REGISTRY'. One item, '34ec33654c version1', is highlighted. Other items include 'hello', '90d350d688', and 'us.gcr.io/tarkshyas-1636464439420/github\_nandiniiks3\_temp'. Buttons for 'SELECT' and 'CANCEL' are at the bottom of the modal.

This screenshot shows the same service creation interface, but with different values entered. The 'Service name' field now contains 'cloudweb'. The 'Region' dropdown is still set to 'us-central1 (Iowa)'. The rest of the interface remains largely the same, including the container image selection dialog on the right which is now closed.

Set the container port as 80.

The screenshot shows the Google Cloud Platform Cloud Run service creation interface. The top navigation bar includes tabs for Subscriptions, Images, Instances, Cloud Run, Cloud Build, Get started, website, What's New, Predictions, SQL, and a plus sign. Below the navigation bar, a banner encourages starting a free trial with \$300 in credit. The main header is "Google Cloud Platform" with a dropdown for "tarkshyas-1636464439420". A search bar says "Search products and resources". The left sidebar has "Cloud Run" selected and a "Create service" button. The main content area is titled "Container, Variables & Secrets, Connections, Security". The "CONTAINER" tab is active, showing the "General" section. Under "General", the "Container port" is set to 80. A note states: "Requests will be sent to the container on this port. We recommend listening on \$PORT instead of this specific number." Below this are fields for "Container command" (blank) and "Container arguments" (blank). The "Capacity" section shows "Memory: 512 MB" and "CPU: 1". The taskbar at the bottom shows various application icons.

This screenshot continues from the previous one, showing the "Service settings" and "Configure how this service is triggered" sections. The "Service settings" section is expanded, showing "Ingress" options: "Allow all traffic" (selected), "Allow internal traffic and traffic from Cloud Load Balancing", and "Allow internal traffic only". The "Authentication" section is also expanded, showing "Allow unauthenticated invocations" (selected) and "Require authentication". The "Trigger" tab is visible at the bottom. The taskbar at the bottom shows various application icons.

Cloud run service created successfully.

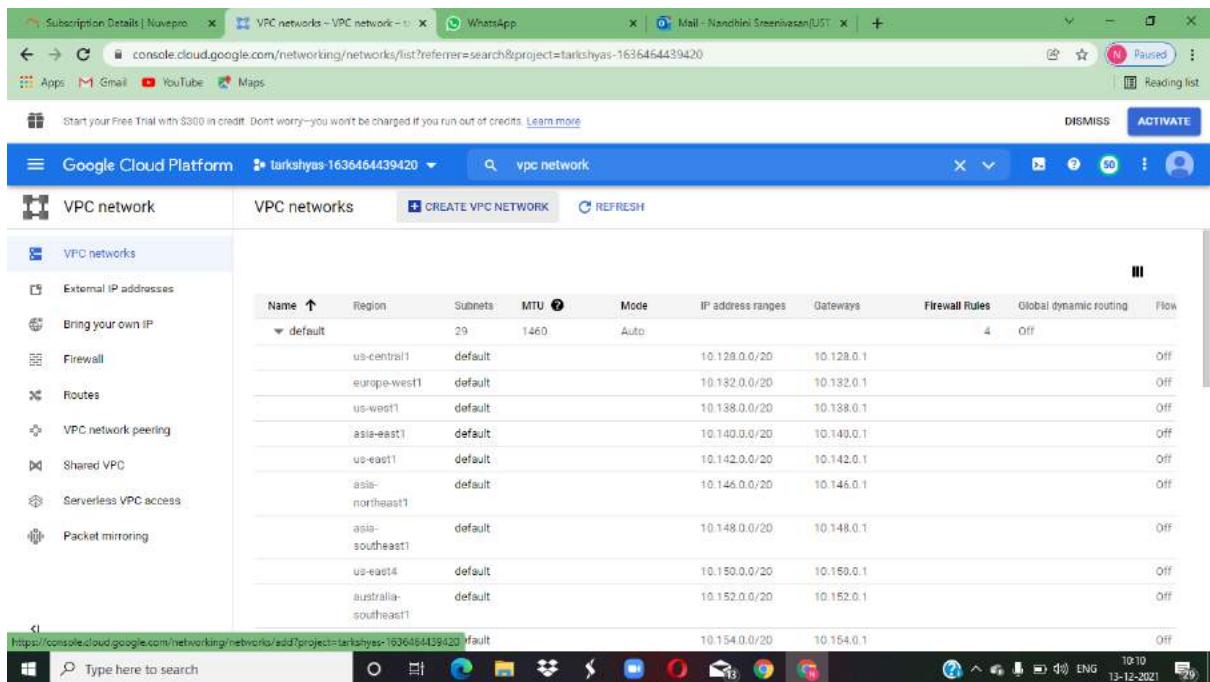
The screenshot shows the Google Cloud Platform Cloud Run service list. A single service, 'cloudweb', is listed. It has 0 requests per second, is in the us-central1 region, uses unauthenticated authentication, and has an ingress of All. It was last deployed just now by 'tsgcp\_user03@tarkhyas'. A modal window at the bottom center says 'Service deleted'. The status bar at the bottom right shows the date as 13-12-2021 and time as 16:55.

Click on the url to view the website.

The screenshot shows a web browser displaying a website for 'Above.'. The main heading is 'School Education' with the subtext 'Get all courses with on-line content'. The background image features a smiling graduate in a cap and gown. The status bar at the bottom right shows the date as 13-12-2021 and time as 16:55.

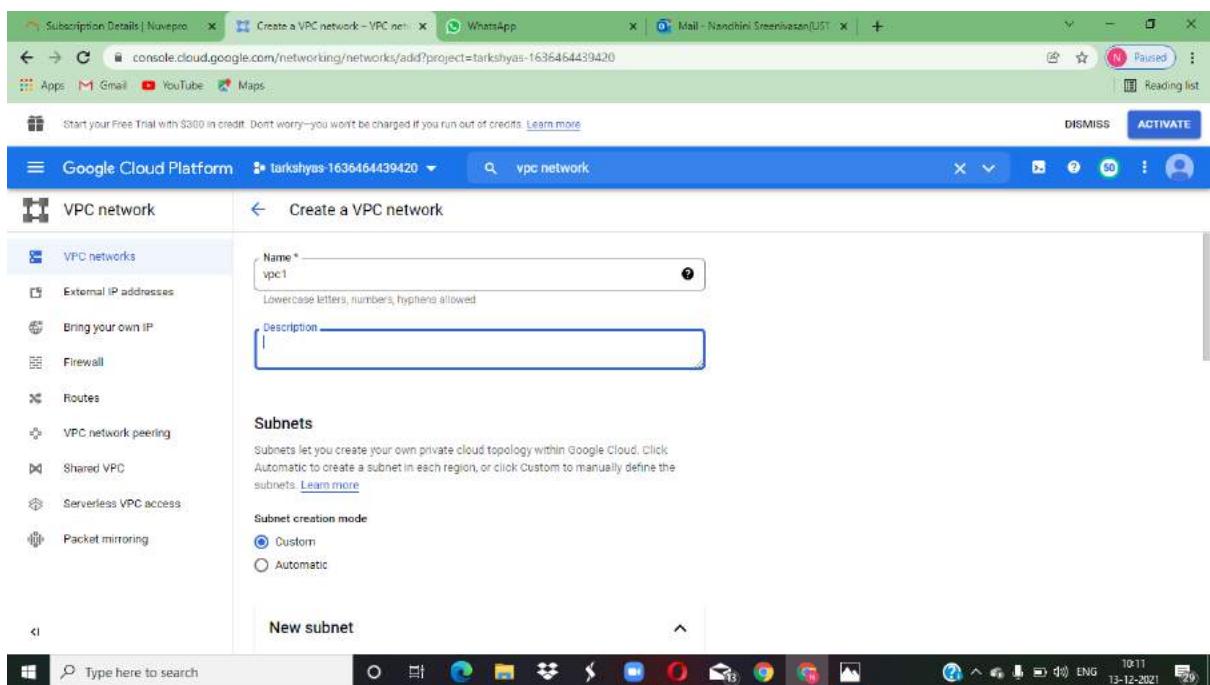
## 7. Create VPC peering to communicate between two VM instances in different networks.

Click on Create VPC Network.



The screenshot shows the Google Cloud Platform interface for managing VPC networks. The left sidebar lists various options like VPC networks, External IP addresses, Firewall, Routes, VPC network peering, Shared VPC, Serverless VPC access, and Packet mirroring. The main content area displays a table of existing VPC networks. The columns include Name, Region, Subnets, MTU, Mode, IP address ranges, Gateways, Firewall Rules, Global dynamic routing, and Flow. A 'CREATE VPC NETWORK' button is located at the top right of the main content area.

Name	Region	Subnets	MTU	Mode	IP address ranges	Gateways	Firewall Rules	Global dynamic routing	Flow
default	us-central1	default	1460	Auto	10.128.0.0/20	10.128.0.1	4	Off	Off
	europe-west1	default			10.132.0.0/20	10.132.0.1			Off
	us-west1	default			10.138.0.0/20	10.138.0.1			Off
	asia-east1	default			10.140.0.0/20	10.140.0.1			Off
	us-east1	default			10.142.0.0/20	10.142.0.1			Off
	asia-northeast1	default			10.146.0.0/20	10.146.0.1			Off
	asia-southeast1	default			10.148.0.0/20	10.148.0.1			Off
	us-east4	default			10.150.0.0/20	10.150.0.1			Off
	australia-southeast1	default			10.152.0.0/20	10.152.0.1			Off



The screenshot shows the 'Create a VPC network' page. The left sidebar is identical to the previous screen. The main area has a 'Name' field set to 'vpc1' with a note about lowercase letters, numbers, and hyphens. There is a 'Description' field with a single character. Below these are sections for 'Subnets' and 'Subnet creation mode'. Under 'Subnet creation mode', the 'Custom' option is selected. At the bottom, there is a 'New subnet' button.

Create subset in custom mode.

The screenshot shows the 'Create a VPC network' wizard in the Google Cloud Platform. The left sidebar lists options like VPC networks, External IP addresses, Bring your own IP, Firewall, Routes, VPC network peering, Shared VPC, Serverless VPC access, and Packet mirroring. The main panel is titled 'Create a VPC network' and shows 'Subnet creation mode' set to 'Custom'. A 'New subnet' section is displayed with the following fields:

- Name \*: subnet1
- Description: (empty)
- Region \*: asia-south2
- IP address range \*: 10.2.0.0/24

At the bottom of the panel are 'CREATE SECONDARY IP RANGE' and 'CREATE' buttons.

This screenshot shows the same 'Create a VPC network' wizard after adding a subnet. The 'Subnet creation mode' is now set to 'Automatic'. The main panel shows the newly created subnet 'subnet1' in the 'ADD SUBNET' dropdown. Other settings visible include:

- Dynamic routing mode: Regional (selected)
- Enable DNS API to pick a DNS policy: ENABLE
- Maximum transmission unit (MTU): 1460

At the bottom are 'CREATE' and 'CANCEL' buttons, and an 'EQUIVALENT COMMAND LINE' dropdown.

Click on create VPC Network to create another VPC Network.

Two VPC Networks are created `vpc1` and `vpc2` respectively.

The screenshot shows the Google Cloud Platform VPC networks page. The left sidebar lists various network-related options: VPC networks, External IP addresses, Bring your own IP, Firewall, Routes, VPC network peering, Shared VPC, Serverless VPC access, and Packet mirroring. The main table displays the following VPC networks:

VPC network	Region/Subnet	Name	IP Range	State	
VPC networks	us-west1	default	10.180.0.0/20	10.180.0.1	
	us-west1	default	10.182.0.0/20	10.182.0.1	
	asia-southeast2	default	10.184.0.0/20	10.184.0.1	
	europe-central2	default	10.186.0.0/20	10.186.0.1	
	northamerica-northeast2	default	10.188.0.0/20	10.188.0.1	
	asia-south2	default	10.190.0.0/20	10.190.0.1	
	australia-southeast2	default	10.192.0.0/20	10.192.0.1	
	southamerica-west1	default	10.194.0.0/20	10.194.0.1	
	vpc1	1	1460	Custom	0 Off
		asia-south2	subnet1	10.2.0.0/24	10.2.0.1
vpc2	1	1460	Custom	0 Off	
	us-east1	subnet2	10.128.0.0/24	10.128.0.1	

For creating External ip addresses for these two vpc networks click on Reserve static address

The screenshot shows the 'Reserve a static address' page for the 'vpc1' VPC network. The left sidebar is identical to the previous screenshot. The main form fields are:

- Name \***: vpc1ext
- Description**: (empty)
- Network Service Tier**: Premium (Current project-level tier, change)
- IP version**: IPv4
- Type**: Regional
- Region**: us-central1 (Iowa)

The screenshot shows the Google Cloud Platform VPC network interface. On the left, a sidebar lists options like VPC networks, External IP addresses, Firewall, Routes, VPC network peering, Shared VPC, Serverless VPC access, and Packet mirroring. The main area is titled 'External IP addresses' and shows a table with one row: 'vpc1ext' (34.181.142.58) in the 'asia-south2' region, type 'Static', version 'IPv4', and no 'In use by'. A success message at the bottom says 'Successfully created address "vpc1ext".' The status bar at the bottom right shows the date as 13-12-2021 and the time as 10:16.

Reserve static address for second vpc network.

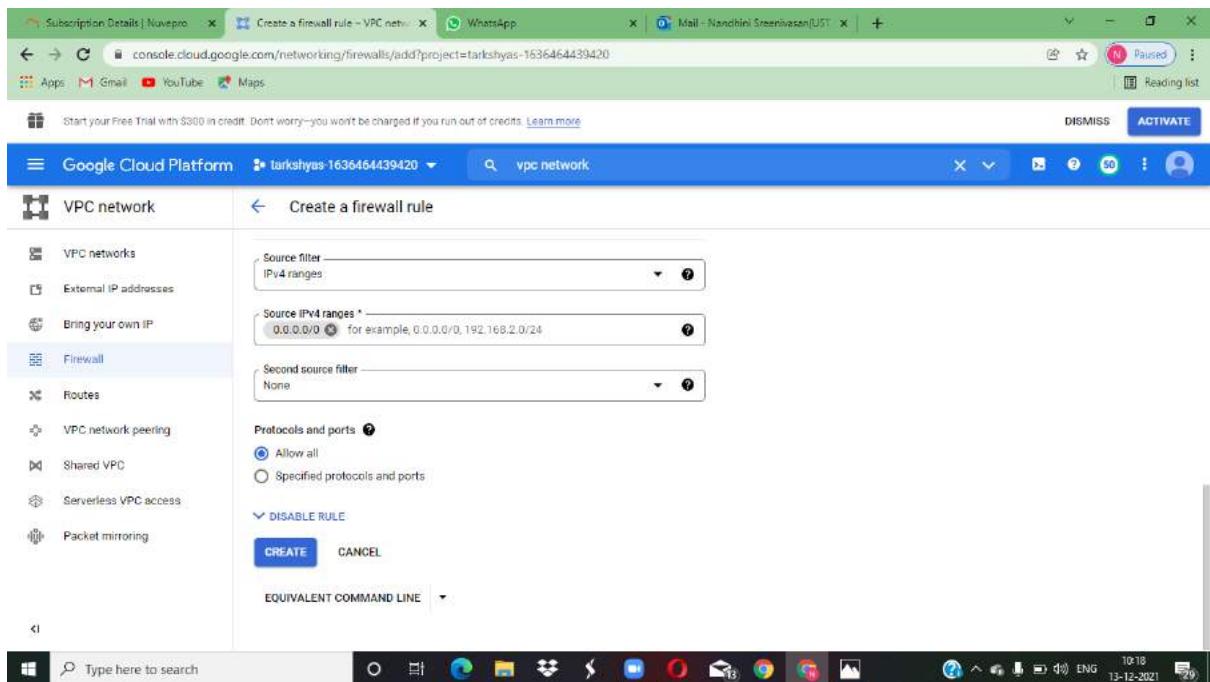
Two external addresses are reserved.

This screenshot is identical to the previous one, showing the creation of a second static external IP address, 'vpc2ext'. The table now contains two rows: 'vpc1ext' and 'vpc2ext'. Both are in the 'asia-south2' region, type 'Static', version 'IPv4', and have 'None' in the 'In use by' column. A success message at the bottom says 'Successfully created address "vpc2ext".' The status bar at the bottom right shows the date as 13-12-2021 and the time as 10:17.

Create Firewall rules for these two vpc networks. Click on Create firewall rule.

The screenshot shows the Google Cloud Platform interface for managing VPC networks. The left sidebar is collapsed, and the main area is titled "Firewall". A sub-section titled "Firewall policies inherited by this project" shows a table with one row: "default-allow-icmp". Below it, a section titled "Firewall rules in this project" also shows a table with one row: "default-allow-icmp". A modal window is open in the center, displaying the message "Successfully created address 'vpcext'." The status bar at the bottom indicates the URL as "https://console.cloud.google.com/networking/firewalls/add?project=tarkshyas-1636464439420" and the time as "10:17 13-12-2021".

The screenshot shows the "Create a firewall rule" page. The left sidebar is collapsed, and the main area has a title bar "Create a firewall rule". The "Name" field is filled with "rule1". The "Logs" section has the "off" radio button selected. The "Network" dropdown is set to "vpc1". The "Priority" field is set to "1000". The status bar at the bottom indicates the URL as "https://console.cloud.google.com/networking/firewalls/add?project=tarkshyas-1636464439420" and the time as "10:18 13-12-2021".



Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. <a href="#">Learn more</a>									
<a href="#">CREATE FIREWALL RULE</a> <a href="#">REFRESH</a> <a href="#">CONFIGURE LOGS</a> <a href="#">DELETE</a>									
<input type="checkbox"/> <a href="#">Filter</a> Enter property name or value									
Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network	Logs	
default-allow-icmp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	icmp	Allow	65534	default	Off	<input checked="" type="checkbox"/>
default-allow-internal	Ingress	Apply to all	IP ranges: 10.0.0.0/16	tcp:0-65535, udp:0-65535, icmp	Allow	65534	default	Off	<input checked="" type="checkbox"/>
default-allow-rdp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:3389	Allow	65534	default	Off	<input checked="" type="checkbox"/>
default-allow-ssh	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:22	Allow	65534	default	Off	<input checked="" type="checkbox"/>
frule1	Ingress	Apply to all	IP ranges: 0.0.0.0/0	all	Allow	1000	vpc1	Off	<input checked="" type="checkbox"/>

Likewise create another firewall rule

Thus two firewall rules are created frule1 and frule2 respectively.

Google Cloud Platform - tarkshyas-1636464439420 - vpc network

**VPC network**

**Firewall**

Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network	Logs
default-allow-icmp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	icmp	Allow	65534	default	Off
default-allow-internal	Ingress	Apply to all	IP ranges: 10.0.0.0/16	tcp:0-65535 udp:0-65535	Allow	65534	default	Off
default-allow-rdp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:3389	Allow	65534	default	Off
default-allow-ssh	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:22	Allow	65534	default	Off
frule1	Ingress	Apply to all	IP ranges: 0.0.0.0/0	all	Allow	1000	vpc1	Off
frule2	Ingress	Apply to all	IP ranges: 0.0.0.0/0	all	Allow	1000	vpc2	Off

For creating VM instances belongs to these two vpc networks click on Compute engine -> VM instances -> Create Instance

Assign same regions as specified in vpc network.

Specify appropriate vpc networks in network field.

Thus created VM instances belongs to vpc1 and vpc2.

Google Cloud Platform - tarkshyas-1636464439420 - VM instances - Compute Engine

**Compute Engine**

**VM instances**

Status	Name	Zone
Up	vpc1inst	asia-south2-a
Up	vpc2inst	us-east1-b

**Select an instance**

Please select at least one resource.

Check whether communication between these two instances belongs to different VPC networks using ping command. And this shows that there is no communication between these two instances.

```
Subscription Details | Naveen x Firewall - VPC network - tarkshya x VM Instances - Compute En... WhatsApp x Mail - Nandini Sreenivasan x + used

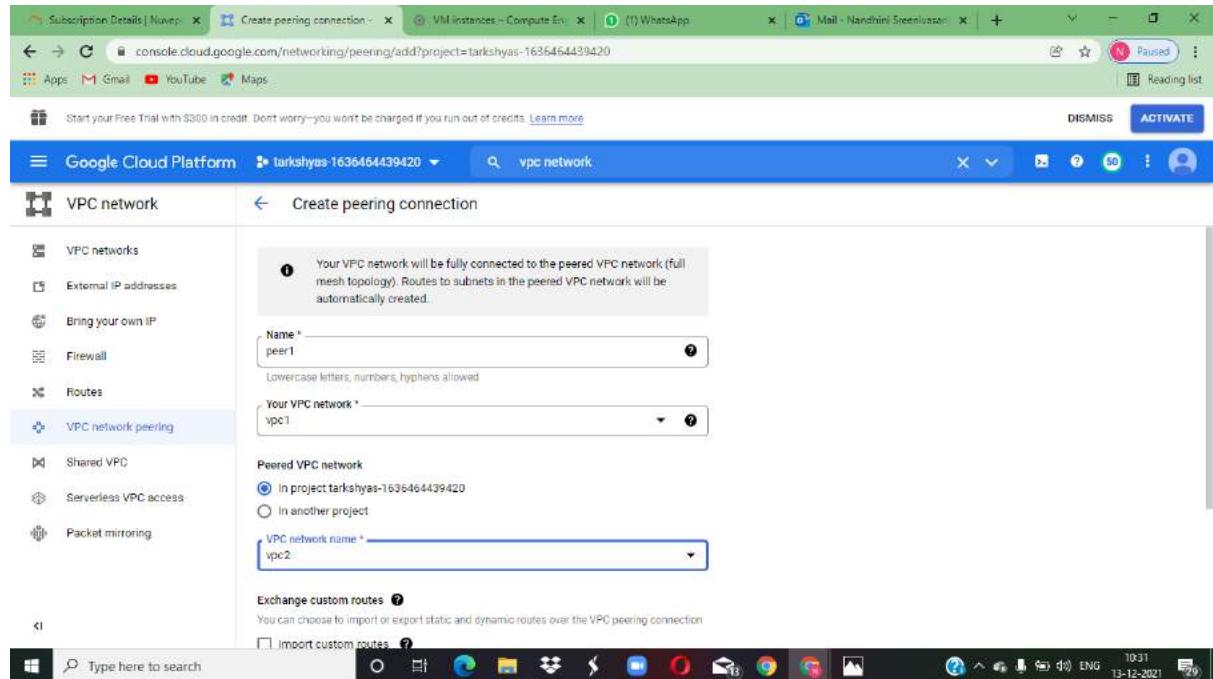
← → C https://console.cloud.google.com/compute/instances?canCreate=true&project=tarkshya tsqip_user03@vpc2inst ~ Google Chrome
ssh.cloud.google.com/projects/tarkshyas-1636464439420/zones/us-east1-b/instances/vpc2inst
Connected, host fingerprint: ssh-rsa 0 3C:5B:7A:62:19:34:ED:04
Connected, host fingerprint: ssh-rsa 0 78:1A:89:7E:1D:AE:1C:67
Connected, host fingerprint: ssh-rsa 0 90:50:5F:9E:06:1C:DC:8C:43:6A:71:8A:5D:69:5E:20:47
Linux vpc2inst 4.19.0-18-cloud-amd64 #1 SMP Debian 4.19.208-1 (2021-08-14)
The programs included with the Debian GNU/Linux system are free software.
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
tsqip_user03@vpc2inst:~$ ping 10.129.0.2
PING 10.129.0.2 (10.129.0.2) 56(84) bytes of data.
tsqip_user03@vpc2inst:~$ ping 10.2.0.2
PING 10.2.0.2 (10.2.0.2) 56(84) bytes of data.
```

To make the communication possible we uses the concept called VPC Peering.

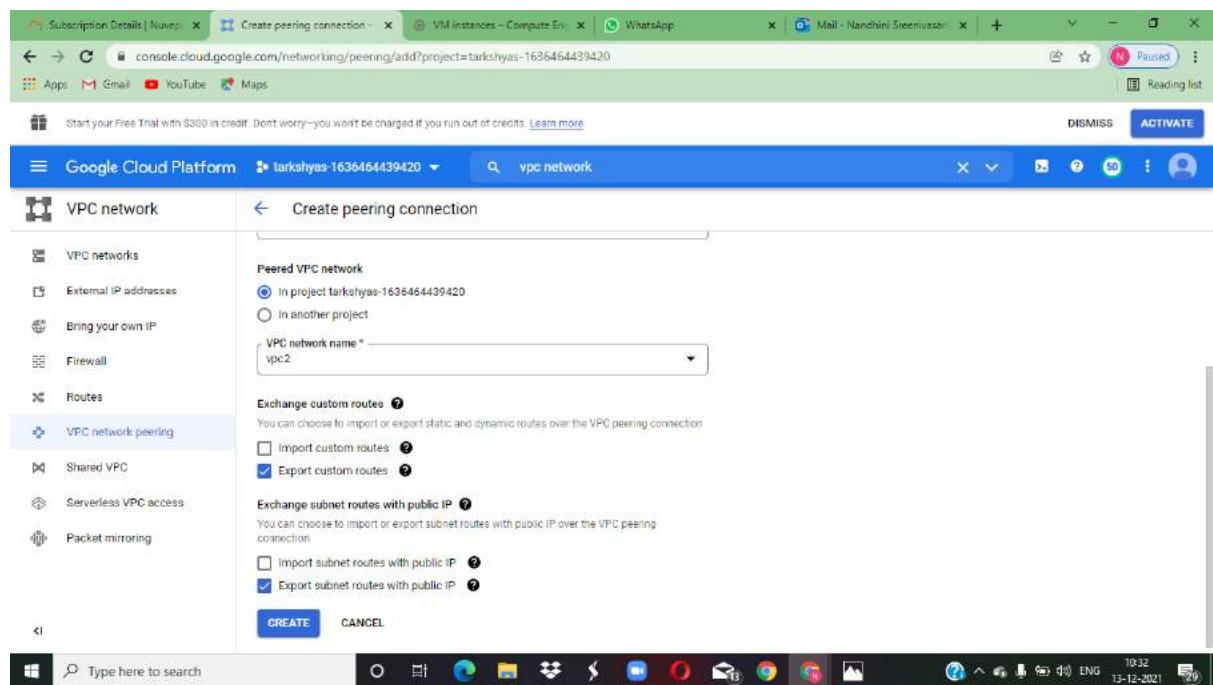
To establish VPC Peering create two peering connections by click on VPC Peering -> Create Connection.

The screenshot shows the Google Cloud Platform interface for managing VPC networks. The left sidebar lists various network-related services: VPC networks, External IP addresses, Bring your own IP, Firewall, Routes, VPC network peering (which is selected and highlighted in blue), Shared VPC, Serverless VPC access, and Packet mirroring. The main content area is titled "VPC network peering" and contains a brief description: "Cloud VPC Network Peering lets you privately connect two VPC networks, which can reduce latency, cost, and increase security. To get started click 'Create connection'." Below this is a "CREATE CONNECTION" button. At the top of the page, there are several tabs: "Subscription Details | Nuvve", "VPC network peering - VPC", "VM instances - Compute Env", "WhatsApp", "Mail - Nandini Sreenivasan", and "Paused". The address bar shows the URL: "console.cloud.google.com/networking/peering/list?project=tarkshyas-1636464439420". The top right corner includes a "DISMISS" button, a "Reading list" icon, and a user profile icon.

Enter name of peering connection. Under the vpc network select a network you want to peer. Select the network to peer with.



Export custom routes and click create.



One peer connection is established.

Likewise create another peering connection.

The screenshot shows the Google Cloud Platform interface with multiple tabs open. The main window displays the 'VPC network peering' section under the 'VPC network' category. It lists two peering connections: 'peer1' (from project tarkshyas to vpc1) and 'peer2' (from project tarkshyas to vpc2). Both are active and export custom routes. A success message at the bottom right says 'Successfully created peering connection "peer2"'.

Now the communication between these two instances are possible with the use of peer connection.

The screenshot shows a terminal session on a Debian VM instance (vpc2inst). The user runs a ping command to another instance in the same VPC. The output shows the ping packets being sent to 10.128.0.2 (the IP of the target instance) over 238 ms intervals.

```

Connected, host fingerprint: ssh-rsa 0 JC:5B:7A:62:19:35:ED:52:6B:58:87:49:C0:D3:29:99:0B:FD:39:86:28:76:56:62:C7:70:04
Linux vpc2inst 4.19.0-18-cloud-amd64 #1 SMP Debian 4.19.208-1 (2021-09-29)
4

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

tsgcp_user@vpc2inst:~$ ping 10.2.0.2
PING 10.2.0.2 (10.2.0.2) 56(84) bytes of data.
64 bytes from 10.2.0.2: icmp_seq=189 ttl=64 time=237 ms
64 bytes from 10.2.0.2: icmp_seq=190 ttl=64 time=237 ms
64 bytes from 10.2.0.2: icmp_seq=191 ttl=64 time=238 ms
64 bytes from 10.2.0.2: icmp_seq=192 ttl=64 time=238 ms
64 bytes from 10.2.0.2: icmp_seq=193 ttl=64 time=238 ms
64 bytes from 10.2.0.2: icmp_seq=194 ttl=64 time=238 ms
64 bytes from 10.2.0.2: icmp_seq=195 ttl=64 time=238 ms
64 bytes from 10.2.0.2: icmp_seq=196 ttl=64 time=238 ms
64 bytes from 10.2.0.2: icmp_seq=197 ttl=64 time=238 ms
64 bytes from 10.2.0.2: icmp_seq=198 ttl=64 time=238 ms
64 bytes from 10.2.0.2: icmp_seq=199 ttl=64 time=238 ms
64 bytes from 10.2.0.2: icmp_seq=200 ttl=64 time=238 ms
64 bytes from 10.2.0.2: icmp_seq=201 ttl=64 time=238 ms
64 bytes from 10.2.0.2: icmp_seq=202 ttl=64 time=238 ms
64 bytes from 10.2.0.2: icmp_seq=203 ttl=64 time=238 ms
64 bytes from 10.2.0.2: icmp_seq=204 ttl=64 time=238 ms
64 bytes from 10.2.0.2: icmp_seq=205 ttl=64 time=238 ms
64 bytes from 10.2.0.2: icmp_seq=206 ttl=64 time=238 ms
64 bytes from 10.2.0.2: icmp_seq=207 ttl=64 time=238 ms
64 bytes from 10.2.0.2: icmp_seq=208 ttl=64 time=238 ms

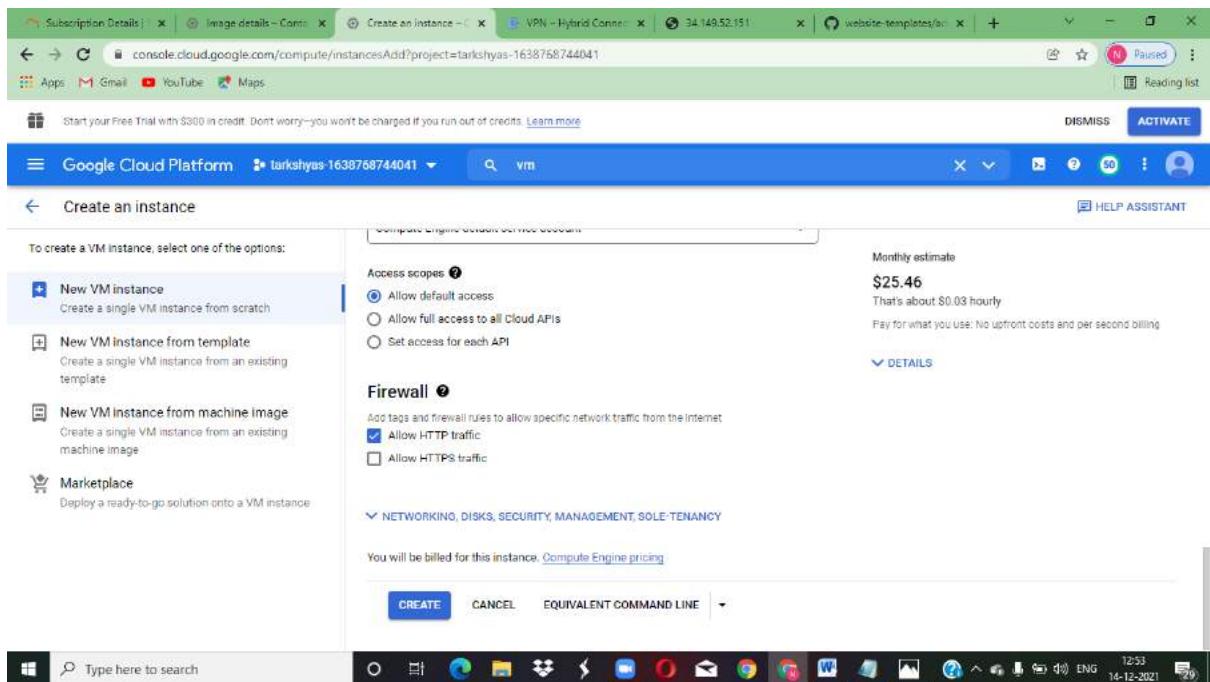
```

## 8. Host a website using Cloud DNS

Create a VM Instance by click on Compute engine -> VM Instances

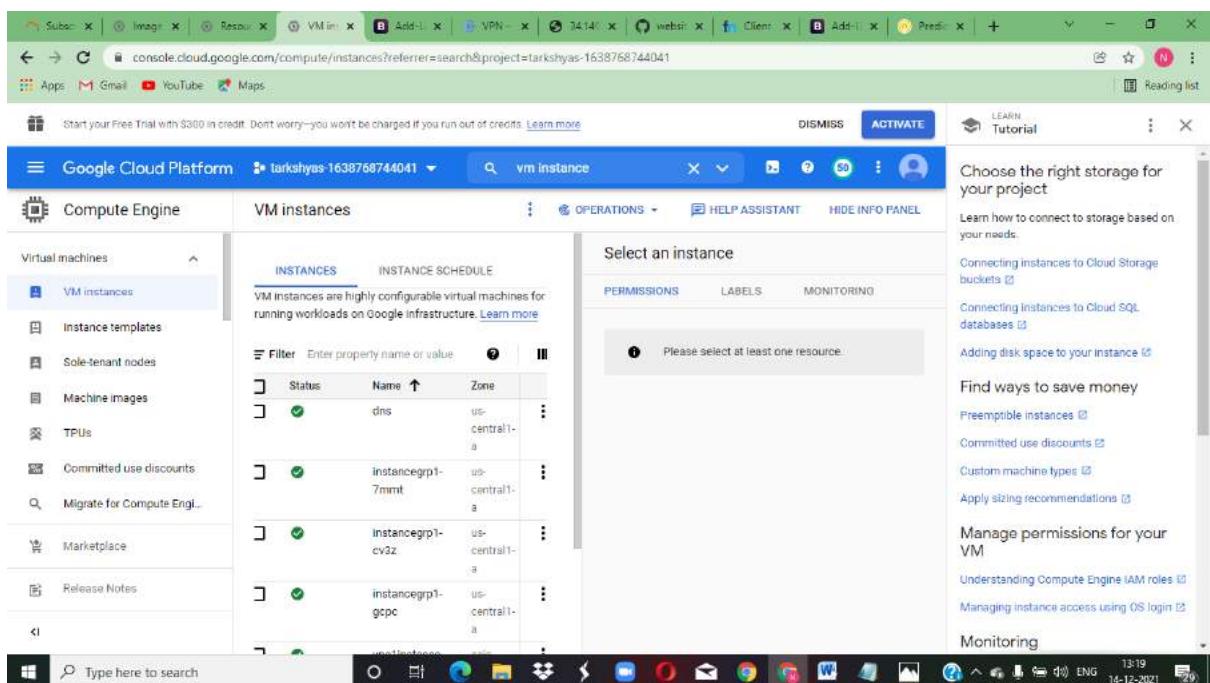
The screenshot shows the Google Cloud Platform Compute Engine interface. On the left, a sidebar lists options like VM instances, Instance templates, Sole-tenant nodes, Machine images, TPUs, Committed use discounts, Migrate for Compute Eng., Marketplace, and Release Notes. The main area displays a list of VM instances with columns for Internal IP, External IP, and SSH. A context menu is open over the fourth instance in the list, containing options such as CREATE INSTANCE, IMPORT VM, REFRESH, START / RESUME, STOP, SUSPEND, RESET, DELETE, and CREATE SCHEDULE. The URL in the browser bar is <https://console.cloud.google.com/compute/instances/Add?project=tarkshyas-1638768744041>.

The screenshot shows the Google Cloud Platform Create an instance page. On the left, a sidebar offers options like New VM instance, New VM instance from template, New VM instance from machine image, and Marketplace. The main area is titled "Configure container" and contains fields for "Container image" (set to `5fd13a14fb19c5d7eeb737b254906c9d9d3e5e2b11d9d9c302c849b61ddcf6`), "Restart policy" (set to "Always"), and checkboxes for "Run as privileged", "Allocate a buffer for STDIN", and "Allocate a pseudo-TTY". It also includes sections for "Command" (with a placeholder), "Arguments" (with a "+ ADD ARGUMENT" button), "Environment variables" (with a "+ ADD VARIABLE" button), and "Volume mounts" (with a "SELECT" button). The URL in the browser bar is <https://console.cloud.google.com/compute/instances/Add?project=tarkshyas-1638768744041>.

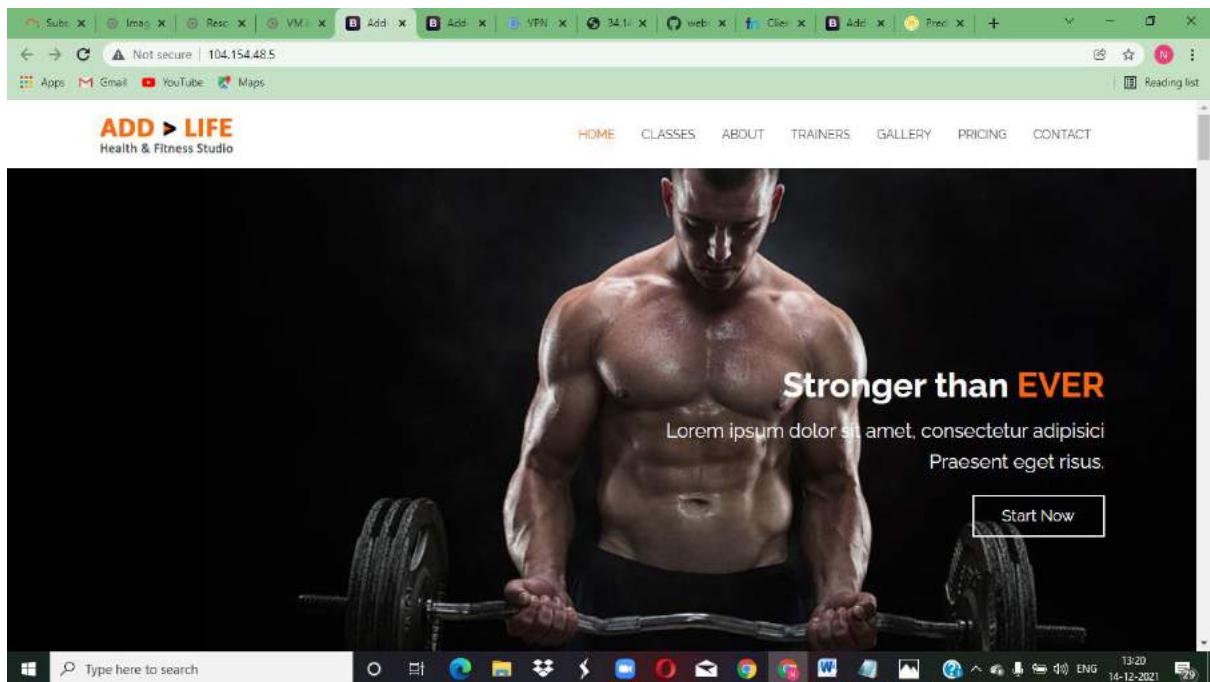


Click create.

VM Instance Created.



Click on external IP.



Create zone by click on Network Service -> Cloud DNS -> Create zone

The screenshot shows the Google Cloud Platform Network Services dashboard. The left sidebar is under 'Network services' and has several options: Load balancing, Cloud DNS (which is selected and highlighted in blue), Cloud CDN, Cloud NAT, Traffic Director, Service Directory, Cloud Domains, Private Service Connect, Marketplace, and Release Notes. The main area is titled 'Cloud DNS' and has a 'CREATE ZONE' button. Below it, there's a section for 'DNS zones' with a sub-section for 'RESPONSE POLICY ZONES'. A message at the top says 'Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. Learn more' with 'DISMISS' and 'ACTIVATE' buttons. To the right, there's an 'Info panel' with the message 'Select a managed zone' and a note about labels. At the bottom, there's a URL bar showing the full URL and a system tray with various icons.

Enter zone name and DNS name.

The screenshot shows the 'Create a DNS zone' page in the Google Cloud Platform Network services section. The URL in the address bar is `console.cloud.google.com/net-services/dns/zones/new/create?project=tarkshyas-1636464439420`. The page title is 'Create a DNS zone'. On the left, there's a sidebar with options like Load balancing, Cloud DNS (which is selected), Cloud CDN, Cloud NAT, Traffic Director, Service Directory, Cloud Domains, Private Service Connect, Marketplace, and Release Notes. The main form has fields for 'Zone name' (set to 'tarkshya03.gq'), 'DNS name' (set to 'tarkshya03.gq'), 'DNSSEC' (set to 'Off'), and a 'Description' field. At the bottom are 'CREATE' and 'CANCEL' buttons.

This screenshot is identical to the one above, but it includes an additional 'Cloud Logging' section at the bottom of the form. It contains radio buttons for 'On' and 'Off', with 'Off' selected. Below this, there's a note: 'After creating your zone, you can add resource record sets and modify the networks your zone is visible on.'

Thus DNS zone created.

The screenshot shows the Google Cloud Platform Network services - Zone details page for the zone 'tarkshya03-gq'. The left sidebar lists various network services: Load balancing, Cloud DNS (selected), Cloud CDN, Cloud NAT, Traffic Director, Service Directory, Cloud Domains, Private Service Connect, Marketplace, and Release Notes. The main pane displays the 'Zone details' for 'tarkshya03-gq'. It shows the DNS name 'tarkshya03.gq.', Type 'Public', DNSSEC 'off', and Cloud Logging 'off'. Below this is a 'RECORD SETS' section with buttons for 'ADD RECORD SET', 'DELETE RECORD SETS', and 'REFRESH'. A table lists record sets: SOA (tarkshya03.gq., TTL 21600, Default) and NS (tarkshya03.gq., TTL 21600, Default). At the bottom is an 'EQUIVALENT REST' section.

Copy these Name servers.

The screenshot shows the Google Cloud Platform Network services - Resource record set details page for the NS type. The left sidebar is identical to the previous screenshot. The main pane shows the 'Resource record set details' for the NS type. It lists the DNS name 'tarkshya03.gq.', Type 'NS', and TTL(seconds) '21600'. Below this is a 'Data' section with a table titled 'Filter' showing four entries: ns-cloud-b1.google.com., ns-cloud-b2.google.com., ns-cloud-b3.google.com., and ns-cloud-b4.google.com.. At the bottom is an 'EQUIVALENT REST' section.

Freenom provides free domain name.

Create one free domain in Freenom.

Enter Domain to Find

Domain	Registration Date	Expiry date	Status	Type
tarkshya03.gq	2021-11-18	2022-02-18	ACTIVE	Free

Results Per Page: 10 1 Records Found. Page 1 of 1

Add the new Name servers.

Information Upgrade Management Tools Manage Freenom DNS

Nameservers

t:	g03.gq	ACTIVE
----	--------	--------

URL Forwarding

Cancel domain

18/02/2022

\* Back to Domains List

The screenshot shows the freenom client area interface. The URL in the address bar is [my.freenom.com/clientarea.php?action=domaindetails](https://my.freenom.com/clientarea.php?action=domaindetails). The main title is "Managing tarkshya03.gq". Below it, there are tabs for "Information", "Upgrade", "Management Tools", and "Manage Freenom DNS". A blue banner at the top says "Changes Saved Successfully!". On the left, there's a sidebar titled "Information" with a sub-section for "Domain details". It shows the domain is "tarkshya03.gq" (ACTIVE), registered on "18/11/2021", and has an "Expiry date". A "Back to Domains List" button is at the bottom of this sidebar.



Create A Record by click on add record set.

The screenshot shows the Google Cloud Platform Network services page. The URL in the address bar is [console.cloud.google.com/net-services/dns/zones/tarkshya03-gq/details?project=tarkshyas-1636464439420](https://console.cloud.google.com/net-services/dns/zones/tarkshya03-gq/details?project=tarkshyas-1636464439420). The main title is "Zone details" for "tarkshya03-gq". There are tabs for "EDIT", "+ ADD RECORD SET", "DELETE ZONE", and "LOGS". A "DISMISS" and "ACTIVATE" button are at the top right. The left sidebar lists "Network services" with "Cloud DNS" selected. The "RECORD SETS" section shows two entries: "tarkshya03.gq." (SOA) and "tarkshya03.gq." (NS). Below this is a table for "ADD RECORD SET" with a "Filter" option. The table has columns for "DNS name", "Type", "TTL (seconds)", and "Routing policy". Two entries are listed: "tarkshya03.gq." (SOA) and "tarkshya03.gq." (NS). At the bottom, there's an "EQUIVALENT REST" link. The Windows taskbar at the bottom shows multiple open browser tabs.

Select record type as A and put the external ip address of the VM instance in the place of IPv4 Address. Click on create.

The screenshot shows the Google Cloud Platform Network services - Create record set page for Cloud DNS. The left sidebar lists various services: Load balancing, Cloud DNS (selected), Cloud CDN, Cloud NAT, Traffic Director, Service Directory, Cloud Domains, Private Service Connect, Marketplace, and Release Notes. The main area is titled "Create record set" and shows the configuration for a new DNS record. The "DNS Name" field contains "tarkshya03.gq.". The "Resource Record Type" dropdown is set to "A", with "TTL" set to "5" and "TTL Unit" set to "minutes". Under "Routing Policy", "Default record type" is selected. The "IPv4 Address" field contains "34.93.85.67". There is a "+ ADD ITEM" button and a "CREATE" button at the bottom.

Create C Record by adding details like record type as CNAME . Click create.

The screenshot shows the Google Cloud Platform Network services - Create record set page for Cloud DNS. The left sidebar is identical to the previous screenshot. The main area is titled "Create record set" and shows the configuration for a new DNS record. The "DNS Name" field contains "www.tarkshya03.gq.". The "Resource Record Type" dropdown is set to "CNAME", with "TTL" set to "5" and "TTL Unit" set to "minutes". Under "Routing Policy", "Default record type" is selected. The "Canonical name" field contains "tarkshya03.gq". There is a "+ ADD ITEM" button and a "CREATE" button at the bottom.

Google Cloud Platform - tarkshya03.gq

Zone details

cloud d

DISMISS ACTIVATE

Network services

Zone details

EDIT ADD RECORD SET DELETE ZONE LOGS REGISTRAR SETUP

Load balancing

Cloud DNS

Cloud CDN

Cloud NAT

Traffic Director

Service Directory

Cloud Domains

Private Service Connect

Marketplace

Release Notes

DNS name: tarkshya03.gq

Type: Public

DNSSEC: Off

Cloud Logging: Off

RECORD SETS

ADD RECORD SET DELETE RECORD SETS REFRESH

Filter: Filter record sets

DNS name	Type	TTL (seconds)	Routing policy
tarkshya03.gq	A	300	Default
tarkshya03.gq	SOA	21600	Default
tarkshya03.gq	NS	21600	Default
www.tarkshya03.gq	CNAME	300	Default

Now using the DNS name we can access the website.

Not secure | tarkshya03.gq

ADD > LIFE  
Health & Fitness Studio

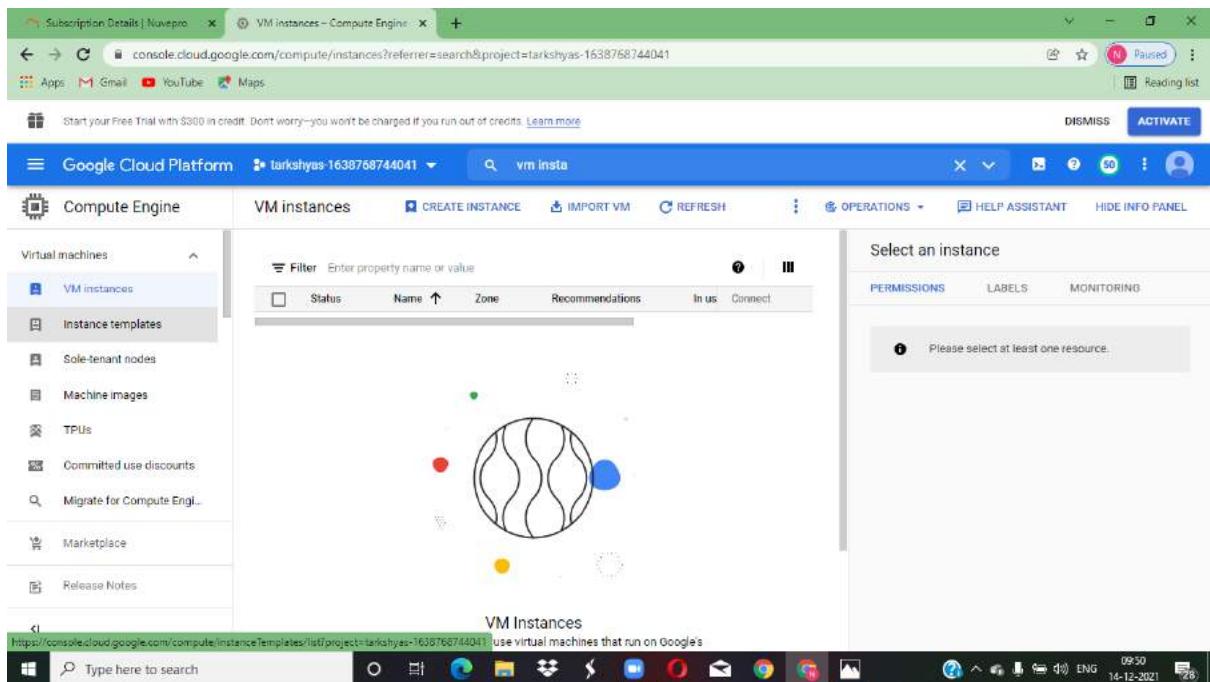
HOME CLASSES ABOUT TRAINERS GALLERY PRICING CONTACT

Stronger than EVER

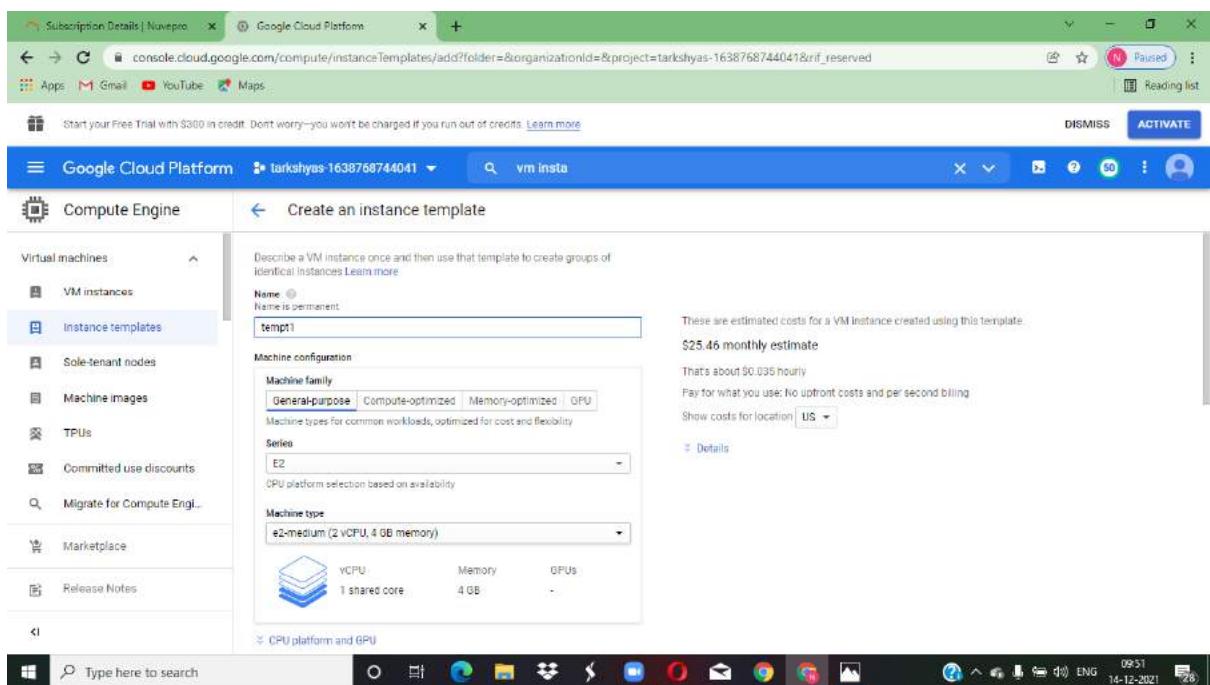
Start Now

## 9. Establish load balancing using HTTP and TCP.

In the cloud console, go to the instance template page.



Name the template as temp1 and boot disk as Debian GNU/Linux 9(stretch).



Enter startup script as follows:

The screenshot shows the 'Create an instance template' page in the Google Cloud Platform Compute Engine interface. On the left, a sidebar lists 'Virtual machines' options: VM instances, Instance templates (which is selected), Sole-tenant nodes, Machine images, TPUs, Committed use discounts, Migrate for Compute Engi.., Marketplace, and Release Notes. The main content area has sections for 'Labels (Optional)', 'Reservations', 'Automation' (with a startup script example), and 'Metadata (Optional)'. At the bottom, there's a toolbar with various icons and a status bar showing '09:54 14-12-2021'.

Create instance group as follows:

The screenshot shows the 'Create an instance group' page in the Google Cloud Platform Compute Engine interface. On the left, a sidebar lists 'To create an instance group, select one of the options': New managed instance group (stateless) and New managed instance group (stateful). The 'New managed instance group (stateless)' option is selected. The main content area shows fields for 'Name' (set to 'instancegrp1'), 'Description (Optional)', 'Location' (set to 'Single zone'), 'Region' (set to 'us-central1 (Iowa)'), 'Zone' (set to 'us-central1-a'), and 'Instance template'. At the bottom, there's a toolbar with various icons and a status bar showing '09:57 14-12-2021'.

Under autoscaling mode, select don't autoscale and specify number of instances as 3.

Subscription Details | Nuvapro Google Cloud Platform

console.cloud.google.com/compute/instanceGroups/add?project=tarkshyas-1638768744041&nif\_reserved

DISMISS ACTIVATE

Google Cloud Platform tarkshyas-1638768744041 vm insta

Create an instance group

New unmanaged instance group

A group of VMs that you manage yourself.

Supports load balancing

Instance template: template1

Number of instances: 1

Autoscaling mode: Don't autoscale

**⚠️** The size of the instance group won't change when autoscaling is turned off. The recommended size will be calculated based on the autoscaling configuration but it won't be applied.

Autoscaling policy: CPU utilization: 60% (default)

Add new metric

Type here to search

Windows taskbar: Start button, File Explorer, Edge, File Manager, Task View, Taskbar icons, ENG 09:58 14-12-2021

This screenshot shows the 'Create an instance group' page in the Google Cloud Platform. It's a 'New unmanaged instance group'. The 'Instance template' is set to 'template1'. There is one instance selected. The 'Autoscaling mode' is set to 'Don't autoscale', which is highlighted with a yellow warning bar stating that the size won't change when autoscaling is turned off. Below this, the 'Autoscaling policy' is set to 'CPU utilization: 60% (default)'. A 'Add new metric' button is available. The bottom of the window has a 'Create' and 'Cancel' button. The Windows taskbar at the bottom shows various pinned and running applications.

Subscription Details | Nuvapro Google Cloud Platform

console.cloud.google.com/compute/instanceGroups/add?project=tarkshyas-1638768744041&nif\_reserved

DISMISS ACTIVATE

Google Cloud Platform tarkshyas-1638768744041 vm insta

Create an instance group

Scale in Controls

Prevent a sudden drop in the number of running VM instances in the group by controlling the process of scaling in. Learn more

Enable Scale in Controls

Delete autoscaling configuration

Less

Autohealing

Health check: No health check

Compute Engine will recreate VM instances only when they're not running.

Advanced creation options

You will be billed for VM instances in this group. Compute Engine pricing

Create Cancel

Equivalent REST or command line

Type here to search

Windows taskbar: Start button, File Explorer, Edge, File Manager, Task View, Taskbar icons, ENG 09:58 14-12-2021

This screenshot shows the same 'Create an instance group' page as the first one, but with more detailed configuration options visible. Under 'Scale in Controls', there is a checkbox for 'Enable Scale in Controls'. Below this, there is a section for 'Autohealing' with a dropdown menu set to 'No health check'. A note says 'Compute Engine will recreate VM instances only when they're not running.' At the bottom, there is a note about billing and a 'Create' and 'Cancel' button. The Windows taskbar at the bottom shows various pinned and running applications.

The screenshot shows the Google Cloud Platform Compute Engine Instance groups page. The left sidebar has 'Virtual machines' expanded, showing 'VM instances', 'Instance templates', and 'Sole-tenant nodes'. The main area displays 'Instance groups' with a table. One row is selected, showing:

Status	Name	Instances	Template	Group type	Creation time	Recommend
Green checkmark	instancegrp1	3	template1	Managed	Dec 14, 2021, 9:58:43 AM UTC+05:30	

The right sidebar contains sections like 'Instance groups', 'How-to guides and tutorials', and 'Autoscaling instance groups'.

## Specify port name mapping

The screenshot shows the 'Edit instancegrp1' page. The left sidebar is identical to the previous screenshot. The main area shows the 'Edit instancegrp1' form with the 'Port name mapping' section highlighted. It shows a table with one entry:

Port name	Port numbers
http	80

Below the table, there's a note: 'The template applies only to newly created VMs. After saving these settings, use this gcloud command to recreate existing VMs using the new template'. The 'Number of instances' field is set to 3.

The screenshot shows the Google Cloud Platform Compute Engine interface. A specific instance group named 'vm insta' is being edited. The configuration includes:

- Virtual machines:** VM instances
- Number of instances:** 3
- Autoscaling:** Enabled, mode: 'Don't autoscale'. A note states: "The size of the instance group won't change when autoscaling is turned off. The recommended size will be calculated based on the autoscaling configuration but it won't be applied."
- Autohealing:** Health check: No health check

At the bottom, there are 'Save' and 'Cancel' buttons.

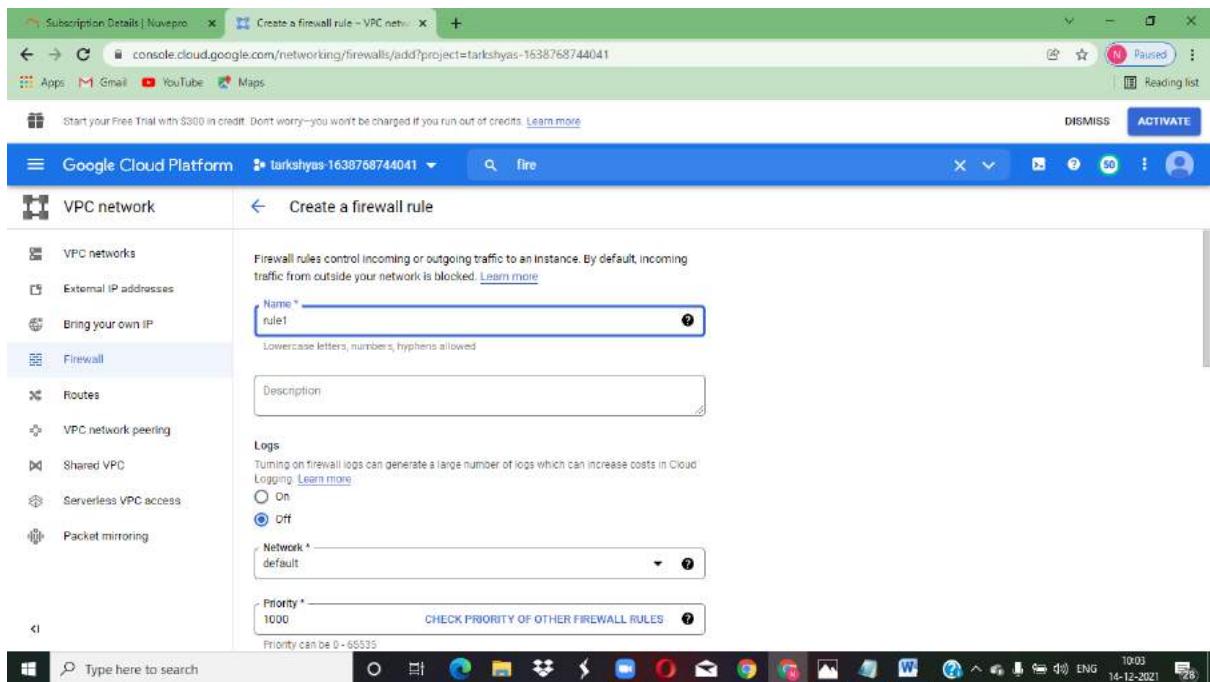
## Configure firewall rule.

The screenshot shows the Google Cloud Platform Networking Firewall rules page. The 'Firewall' section is active, showing:

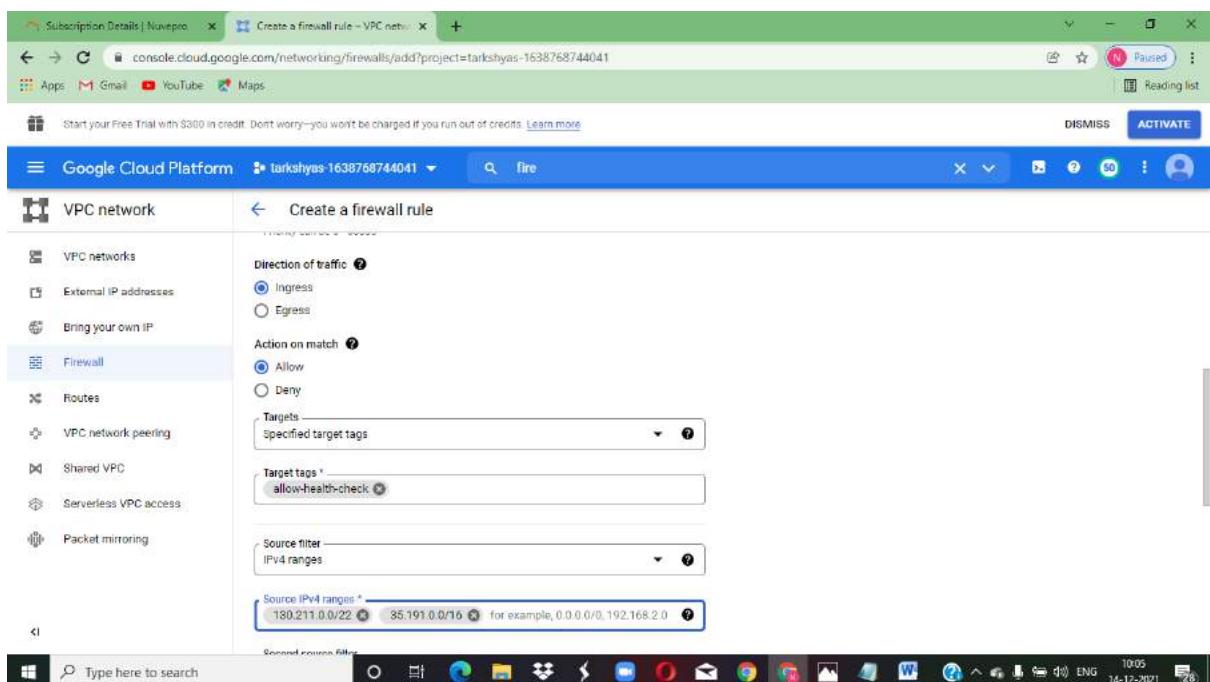
- Firewall policies inherited by this project:** No rows displayed.
- Firewall rules in this project:** One rule named 'default-allow-icmp' is listed. The table details are as follows:

Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network	Logs
default-allow-icmp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	icmp	Allow	65534	default	Off

## Specify name and network .



Under target tags populate allow-health-check and set source IP ranges as following:



Subscription Details | Nuvepro

Create a firewall rule - VPC network

console.cloud.google.com/networking/firewalls/add?project=tarkshyas-1638768744041

DISMISS ACTIVATE

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

Google Cloud Platform tarkshyas-1638768744041 fire

VPC network Create a firewall rule

VPC networks External IP addresses Bring your own IP Firewall Routes VPC network peering Shared VPC Serverless VPC access Packet mirroring

Protocol and ports  Allow all  Specified protocols and ports

tcp: 80

udp: all

Other protocols: protocols, comma separated, e.g. ah, sctp

DISABLE RULE

CREATE CANCEL

EQUIVALENT COMMAND LINE

Type here to search

10:06 14-12-2021 ENG

The screenshot shows the 'Create a firewall rule' page in the Google Cloud Platform. The left sidebar has 'Firewall' selected. The main area shows a single rule: 'tcp: 80'. There's also a section for 'Other protocols' with a placeholder 'protocols, comma separated, e.g. ah, sctp'. At the bottom are 'CREATE' and 'CANCEL' buttons.

Subscription Details | Nuvepro

Firewall - VPC network - tarkshyas

console.cloud.google.com/networking/firewalls/list?referrer=search&project=tarkshyas-1638768744041

DISMISS ACTIVATE

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

Google Cloud Platform tarkshyas-1638768744041 fire

VPC network Firewall CREATE FIREWALL RULE REFRESH CONFIGURE LOGS DELETE

VPC networks External IP addresses Bring your own IP Firewall Routes VPC network peering Shared VPC Serverless VPC access Packet mirroring

Firewall policies inherited by this project

Filter Enter property name or value

Enforcement order	Policy name	Firewall rules	Description	Inherited from	Located at
No rows to display					

Firewall rules in this project

Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. [Learn more](#)

Note: App Engine firewalls are managed in the [App Engine Firewall rules section](#).

Filter Enter property name or value

Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network	Logs
rule1	Ingress	allow-health	IP ranges: 130.1	Tcp:80	Allow	1000	default	Off
default-allow-icmp	Ingress	Apply to all	IP ranges: 0.0.0	Icmp	Allow	65534	default	Off

Type here to search

10:06 14-12-2021 ENG

The screenshot shows the 'Firewall' list page. The left sidebar has 'Firewall' selected. The main area shows two rules: 'rule1' (Ingress, targets 'allow-health', TCP port 80, priority 1000) and 'default-allow-icmp' (Ingress, targets 'Apply to all', ICMP, priority 65534). There are 'CREATE FIREWALL RULE' and 'REFRESH' buttons at the top.

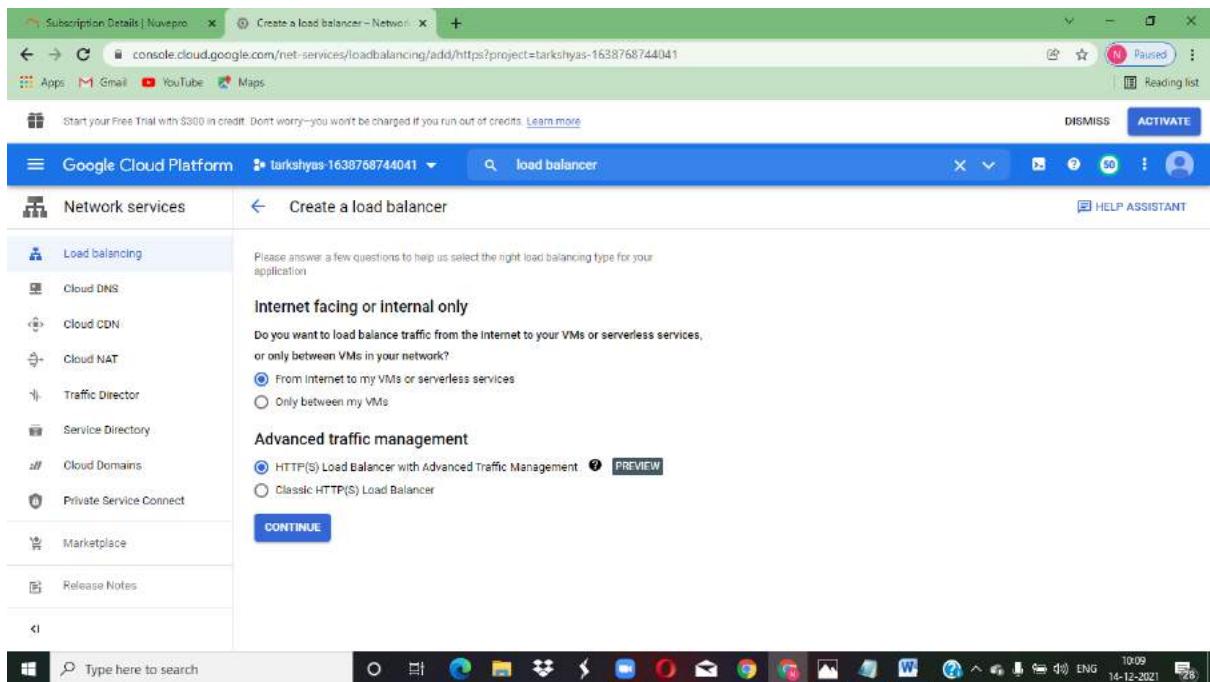
## Reserve external IP addresses

The screenshot shows the Google Cloud Platform VPC network External IP addresses page. A success message at the bottom center says "Successfully created address 'lb-ipv4-1'." The main table lists three IP addresses: one static IPv4 (34.149.52.151) and two ephemeral IPv4s (34.70.188.244 and 35.184.167.54). The interface includes a sidebar with options like VPC networks, Firewall, Routes, VPC network peering, Shared VPC, Serverless VPC access, and Packet mirroring.

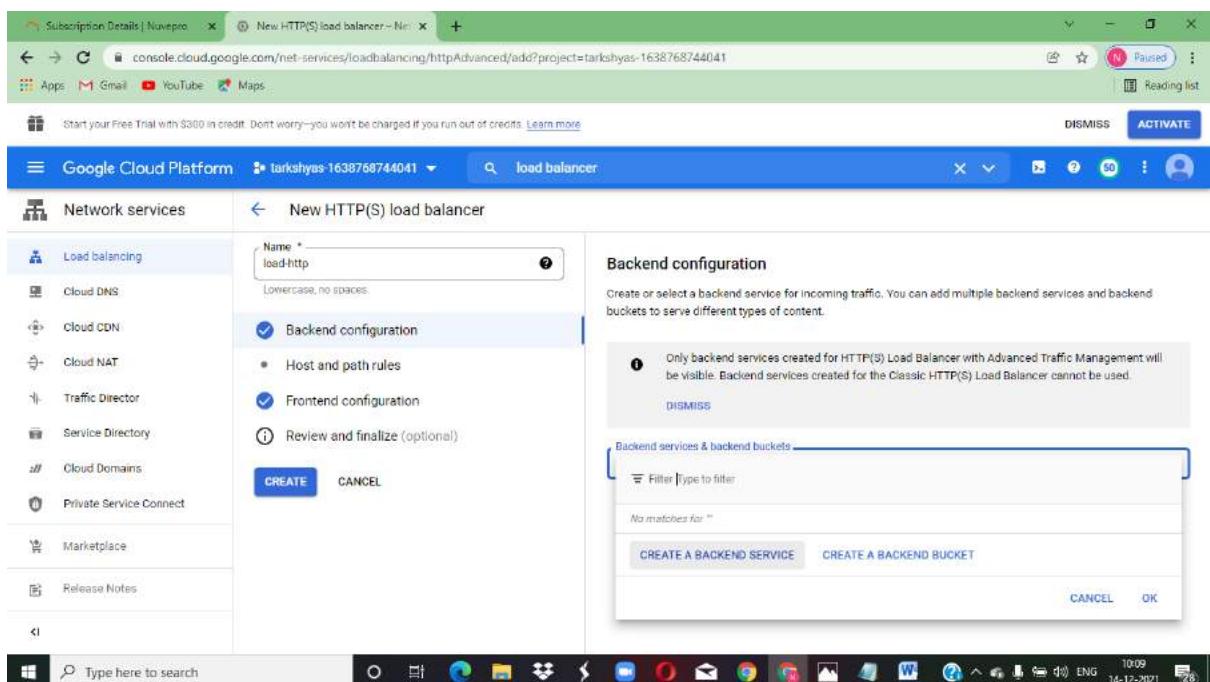
Go to load balancing click create load balancer.

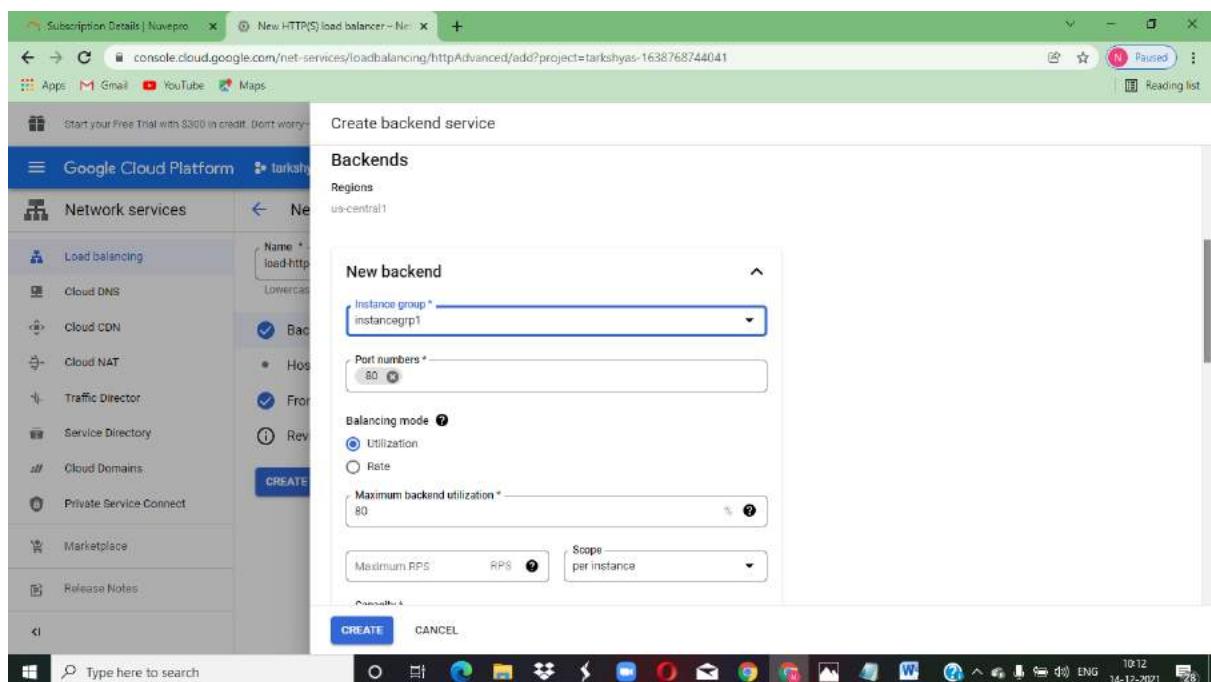
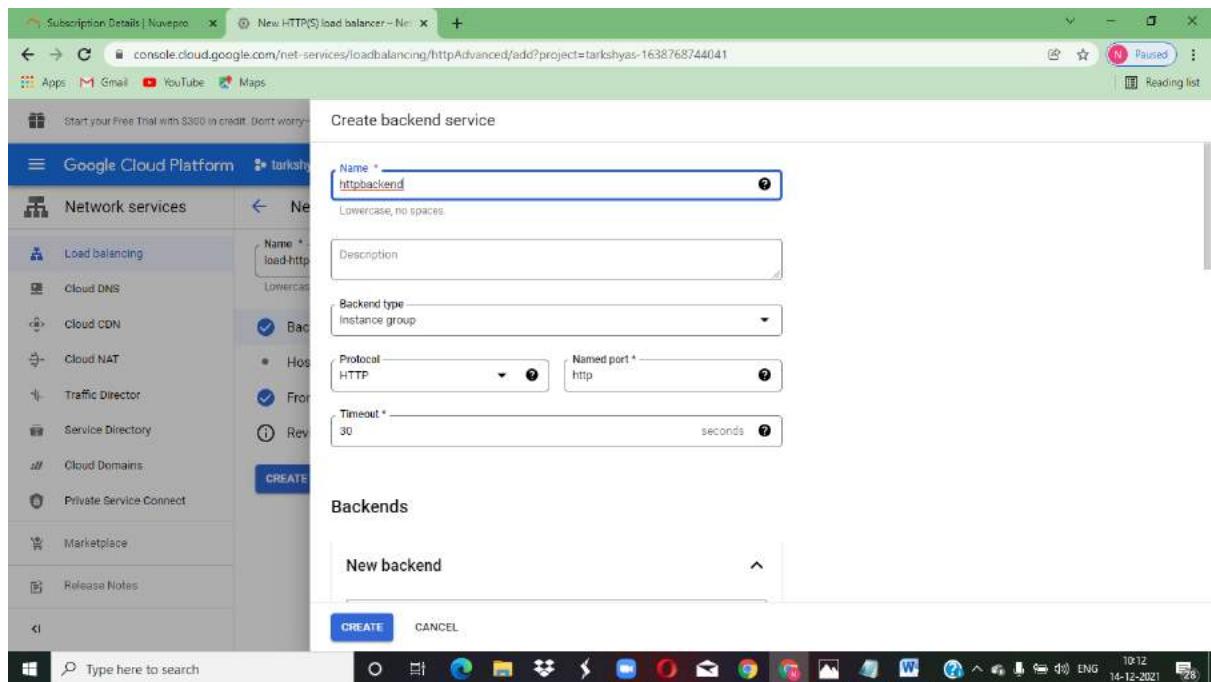
Select http load balancing.

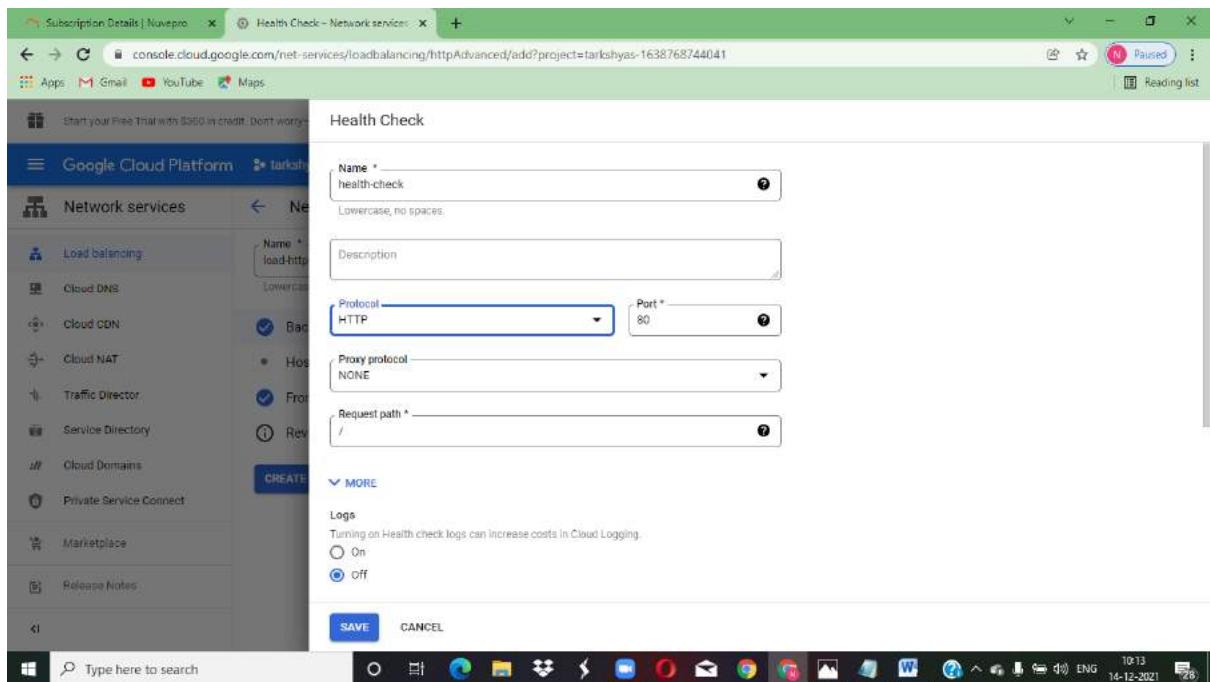
The screenshot shows the Google Cloud Platform Network services Create a load balancer page. The left sidebar under "Load balancing" includes options for Cloud DNS, Cloud CDN, Cloud NAT, Traffic Director, Service Directory, Cloud Domains, Private Service Connect, Marketplace, and Release Notes. The main area displays two configuration panels: "HTTP(S) Load Balancing" and "TCP Load Balancing". Both panels have "Configure" sections for "HTTP LB", "HTTPS LB (includes HTTP/2 LB)", and "TCP Proxy". They also have "Options" sections for "Internet-facing or internal" and "Single or multi-region". Each panel has a "START CONFIGURATION" button at the bottom.



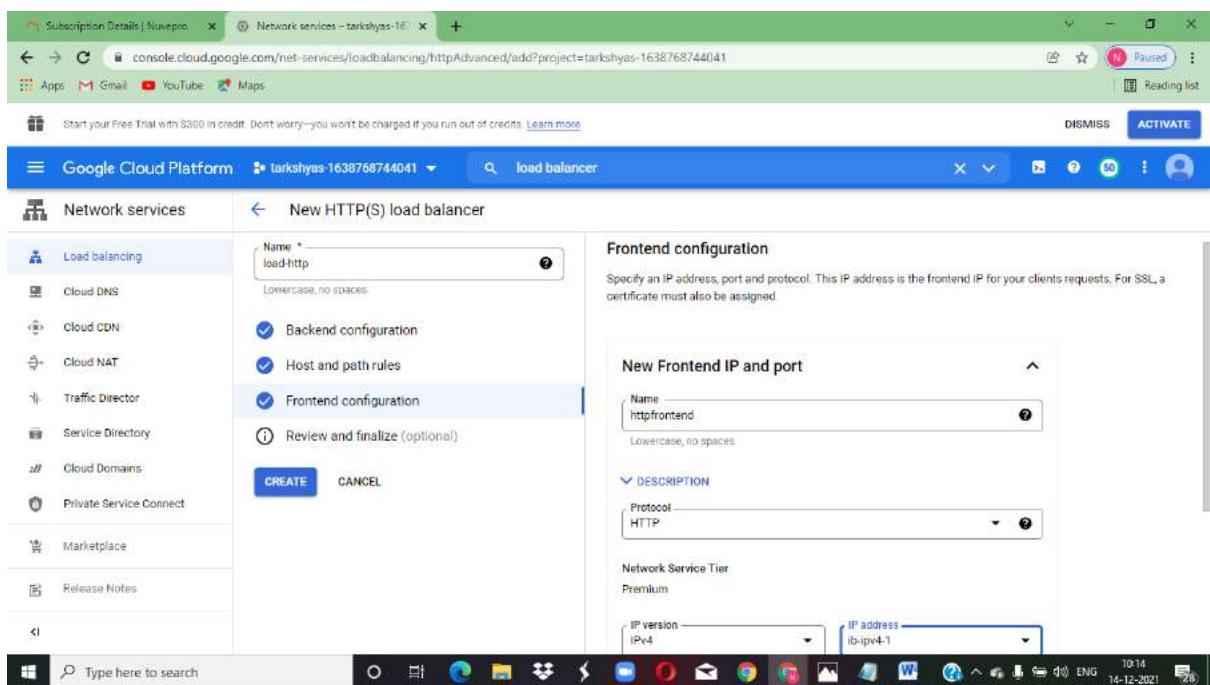
Perform Backend and frontend configuration.







Review and finalize then create.



The screenshot shows the Google Cloud Platform Network services Load Balancing interface. On the left sidebar, under 'Network services', 'Load balancing' is selected. In the main area, a table lists one load balancer:

Name	Load balancer type	Protocols	Region	Backends
load-http	HTTP(S)	HTTP	(Region)	1 backend service (1 instance group, 0 network endpoint groups)

A note below the table says: "To edit load balancing resources like forwarding rules and target proxies, go to the [advanced menu](#)". The status bar at the bottom right shows the date as 14-12-2021.

Copy the IP port address and paste it on the url.

The screenshot shows the 'Load balancer details' page for 'load-http'. The main section displays the configuration for the frontend:

Protocol	IP Port	Certificate	SSL Policy	Network Tier
HTTP	34.149.62.101:80	-	-	Premium

Below this, sections for 'Host and path rules' and 'Backend' are shown. The 'Host and path rules' table has one entry: 'All unmatched (default)'. The 'Backend' section shows a single backend service named 'httpbackend'. The status bar at the bottom right shows the date as 14-12-2021.

These are the 3 instances created.

The screenshot shows the Google Cloud Platform Compute Engine VM instances page. On the left, a sidebar lists 'Virtual machines' with 'VM instances' selected. The main area displays a table of VM instances:

Status	Name ↑	Zone
Running	instancegrp1-7mm	us-central1-a
Running	instancegrp1-cv8z	us-central1-a
Running	instancegrp1-gpc	us-central1-a

A modal window titled 'Select an instance' is open, prompting 'Please select at least one resource.' To the right, a sidebar titled 'Start your project' provides links to 'Connect to your instances', 'Transfer files', 'Find existing VM solutions', 'Explore Marketplace', 'Quickly learn how to build a two-tier web app', 'How-to guides and tutorials', 'Setting up your website host name', 'Reserving a static external IP address', 'Creating and managing Windows instances', 'Explore tools and APIs', 'Learn about the Cloud Shell CLI', and 'Connecting via client libraries'.

Now we can see the multi[ple requests handling by load balancer.

The screenshot shows a browser window with the URL 'Not secure | 34.149.52.151'. The title bar indicates the IP address '34.149.52.151'. The content of the browser shows the text 'Traffic from instancegrp1-7mm'.



Traffic from instancegrp1-cv3z



Traffic from instancegrp1-gcpc



Load Balancing using TCP

The screenshot shows the Google Cloud Platform Network services interface for creating a load balancer. The left sidebar lists various network services: Load balancing, Cloud DNS, Cloud CDN, Cloud NAT, Traffic Director, Service Directory, Cloud Domains, Private Service Connect, Marketplace, and Release Notes. The main content area is titled "Create a load balancer". It displays two main sections: "HTTP(S) Load Balancing" and "TCP Load Balancing". Both sections provide configuration options and "START CONFIGURATION" buttons. A status bar at the bottom indicates "Created sql" and the date "14-12-2021".

This screenshot continues the "Create a load balancer" process. It asks whether the load balancer will be "Internet facing or internal only". The user has selected "From Internet to my VMs". It then asks if the backends will be in "Multiple regions or single region", with "Multiple regions (or not sure yet)" selected. A "CONTINUE" button is visible at the bottom.

The screenshot shows the 'New TCP/SSL load balancer' configuration page in the Google Cloud Platform Network services section. The left sidebar lists various network services: Load balancing, Cloud DNS, Cloud CDN, Cloud NAT, Traffic Director, Service Directory, Cloud Domains, Private Service Connect, Marketplace, and Release Notes. The main panel has a title 'New TCP/SSL load balancer'. It includes fields for 'Name' (tcpload) and 'Timeout' (30 seconds). A sidebar on the right shows 'Backend configuration' selected, with options for 'Frontend configuration' and 'Review and finalize (optional)'. Below these are sections for 'Backends' (Regions: us-central1), 'Instance groups' (instancegrp1), and 'Health check' (hc1). A note at the bottom indicates 'port: 80, timeout: 5s, check interval: 5s, unhealthy threshold: 2 attempts'. The status bar at the bottom shows the date as 14-12-2021 and time as 14:31.

This screenshot is identical to the one above, but the 'Frontend configuration' option is now selected in the sidebar. The main panel displays the 'Frontend configuration' details, which specify an IP address, port, and protocol. A note below states: 'Specify an IP address, port and protocol. This IP address is the frontend IP for your clients requests. For SSL, a certificate must also be assigned.' The rest of the interface, including the sidebar and status bar, remains the same.

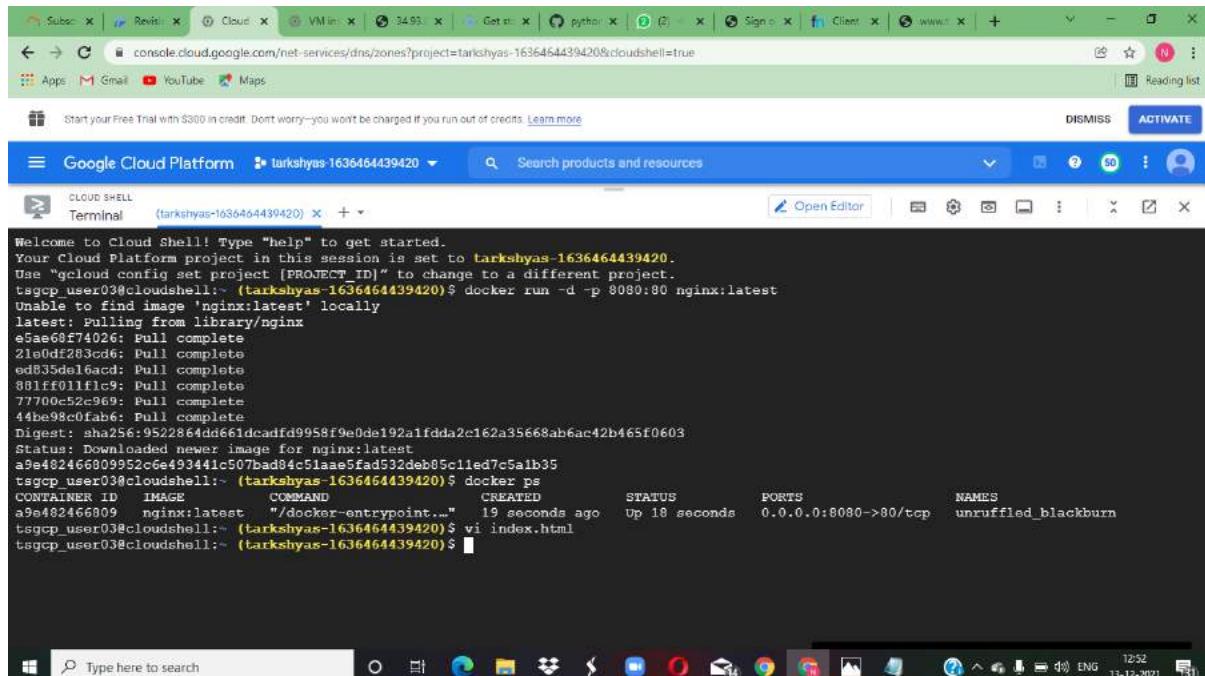
The screenshot shows the Google Cloud Platform Network services Load balancing page. The left sidebar lists various network services: Load balancing, Cloud DNS, Cloud CDN, Cloud NAT, Traffic Director, Service Directory, Cloud Domains, Private Service Connect, Marketplace, and Release Notes. The main content area is titled 'Load balancing' and shows a table for 'LOAD BALANCERS'. A single row is present for 'tcpload', which is an SSL/TCP (Proxy) type load balancer. It is associated with a single backend service (1 instance group, 0 network endpoint groups). Below the table, a note says 'To edit load balancing resources like forwarding rules and target proxies, go to the advanced menu.'

The screenshot shows the 'Load balancer details' page for the 'tcpload' load balancer. The left sidebar is identical to the previous screenshot. The main content area is titled 'tcpload' under 'Frontend'. It shows a table with one row: Protocol (TCP), IP/Port (94.117.164.186.25), Certificate (none), SSL Policy (none), Proxy Protocol (Off), and Network Tier (Premium). Under 'Backend', it shows a table with one row: Endpoint protocol (TCP), Named port (tcp), Timeout (30 seconds), and Health check (hc1). At the bottom, there is a section titled 'ADVANCED CONFIGURATIONS' with a table showing one instance group named 'Instance grp1' with 3 healthy instances. The columns include Name, Type, Zone, Healthy, Autoscaling, Balancing mode, Selected ports, and Capacity.

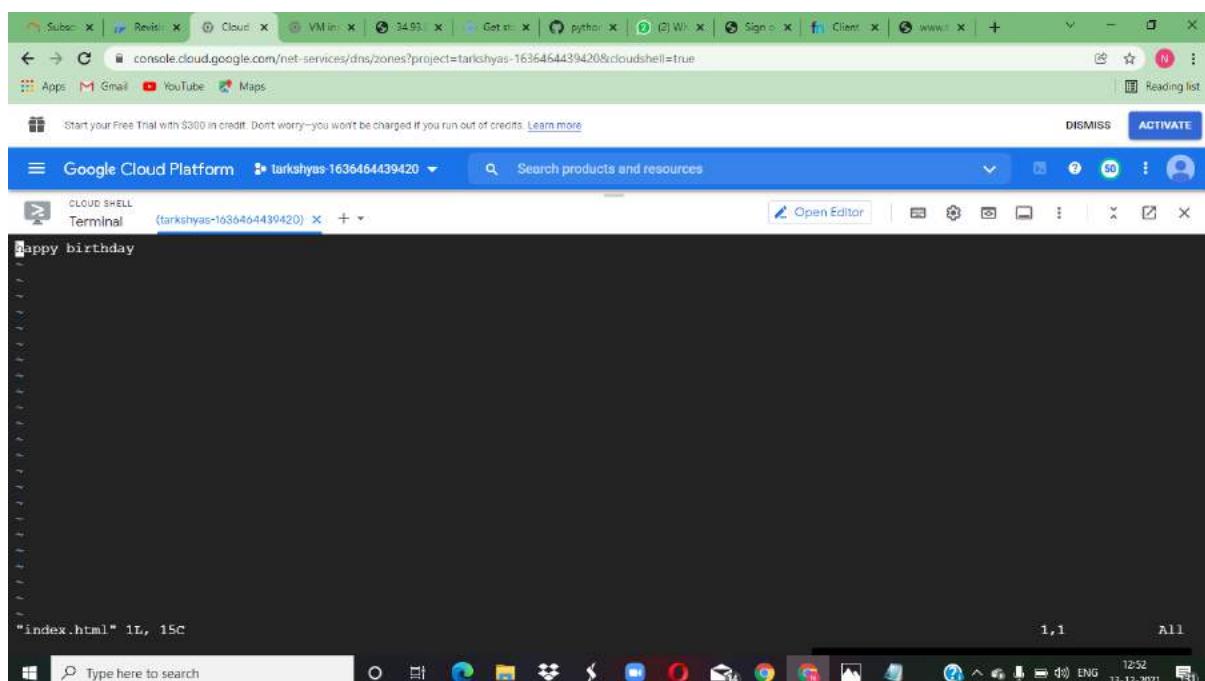
## 10. Create Google Kubernetes Engine to deploy containerized web application.

Activate Cloud shell. Publish your nginx on port 8080 on docker host.

Run the docker ps command to check whether container is running or not.



```
Welcome to Cloud Shell! Type "help" to get started.  
Your Cloud Platform project in this session is set to tarkshyas-1636464439420.  
Use "gcloud config set project [PROJECT_ID]" to change to a different project.  
tsgcp_user@38:~$ docker run -d -p 8080:80 nginx:latest  
Unable to find image 'nginx:latest' locally  
latest: Pulling from library/nginx  
e5ae68f74026: Pull complete  
21e0df283cd6: Pull complete  
ed635de16acd: Pull complete  
881ff011fc9: Pull complete  
77700c52c969: Pull complete  
44be98c0fab6: Pull complete  
Digest: sha256:9522864dd661dcadfd9958f9e0de192a1fdda2c162a35668ab6ac42b465f0603  
Status: Downloaded newer image for nginx:latest  
a9e40246609952c6e493441c507bad04c5laae5fad592deb05c11ed7c5alb35  
tsgcp_user@38:~$ docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
a9e40246609952c6e493441c507bad04c5laae5fad592deb05c11ed7c5alb35  
tsgcp_user@38:~$ docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
a9e40246609952c6e493441c507bad04c5laae5fad592deb05c11ed7c5alb35  
tsgcp_user@38:~$ vi index.html  
tsgcp_user@38:~$
```



```
happy birthday
```

Copy the index.html file to Container.

Welcome to Cloud Shell! Type "help" to get started.  
Your Cloud Platform project in this session is set to **tarkshyas-1636464439420**.  
Use "gcloud config set project [PROJECT\_ID]" to change to a different project.  
tsgcp\_user03@cloudshell:~ (tarkshyas-1636464439420)\$ docker run -d -p 8080:80 nginx:latest  
Unable to find image 'nginx:latest' locally  
latest: Pulling from library/nginx  
e5ae68f74026: Pull complete  
21e0df283cd6: Pull complete  
ed835de16acd: Pull complete  
881ff011fc1c: Pull complete  
77700c52c969: Pull complete  
44be98c0fab6: Pull complete  
Digest: sha256:9522864dd661dcadfd9958f9e0de192a1fdda2c162a35668ab6ac42b465f0603  
Status: Downloaded newer image for nginx:latest  
a9e48246609952c6e493441c507bad84c51aae5fa32deb85c1led7c5alb35  
tsgcp\_user03@cloudshell:~ (tarkshyas-1636464439420)\$ docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
a9e482466009 nginx:latest "/docker-entrypoint..." 19 seconds ago Up 18 seconds 0.0.0.0:8080->80/tcp unruffled\_blackburn  
tsgcp\_user03@cloudshell:~ (tarkshyas-1636464439420)\$ vi index.html  
tsgcp\_user03@cloudshell:~ (tarkshyas-1636464439420)\$ docker cp index.html a9e482466009:/usr/share/nginx/html/  
tsgcp\_user03@cloudshell:~ (tarkshyas-1636464439420)\$ docker commit a9e482466009 cad/web:version1  
sha256:d1b2c9016bbaf26092653d19eac2c479766cc/b961407e6feea00db7c536839  
tsgcp\_user03@cloudshell:~ (tarkshyas-1636464439420)\$

Tag and push the docker image to the repository.

latest: Pulling from library/nginx  
e5ae68f74026: Pull complete  
21e0df283cd6: Pull complete  
ed835de16acd: Pull complete  
881ff011fc1c: Pull complete  
77700c52c969: Pull complete  
44be98c0fab6: Pull complete  
Digest: sha256:9522864dd661dcadfd9958f9e0de192a1fdda2c162a35668ab6ac42b465f0603  
Status: Downloaded newer image for nginx:latest  
a9e48246609952c6e493441c507bad84c51aae5fa32deb85c1led7c5alb35  
tsgcp\_user03@cloudshell:~ (tarkshyas-1636464439420)\$ docker images  
REPOSITORY TAG IMAGE ID CREATED SIZE  
cad/web version1 d1b2c9016bbaf26092653d19eac2c479766cc/b961407e6feea00db7c536839  
nginx latest f652ca396ed1 10 days ago 141MB  
tsgcp\_user03@cloudshell:~ (tarkshyas-1636464439420)\$ docker tag cad/web:version1 us.gcr.io/tarkshyas-1636464439420/cad-site:version1  
tsgcp\_user03@cloudshell:~ (tarkshyas-1636464439420)\$ docker push us.gcr.io/tarkshyas-1636464439420/cad-site:version1

A modal window titled "Authorize Cloud Shell" is displayed, containing the text: "Cloud Shell is requesting your credentials to make a GCP API call. Click to authorize this and future calls that require your credentials." It has two buttons: "REJECT" and "AUTHORIZE".

```

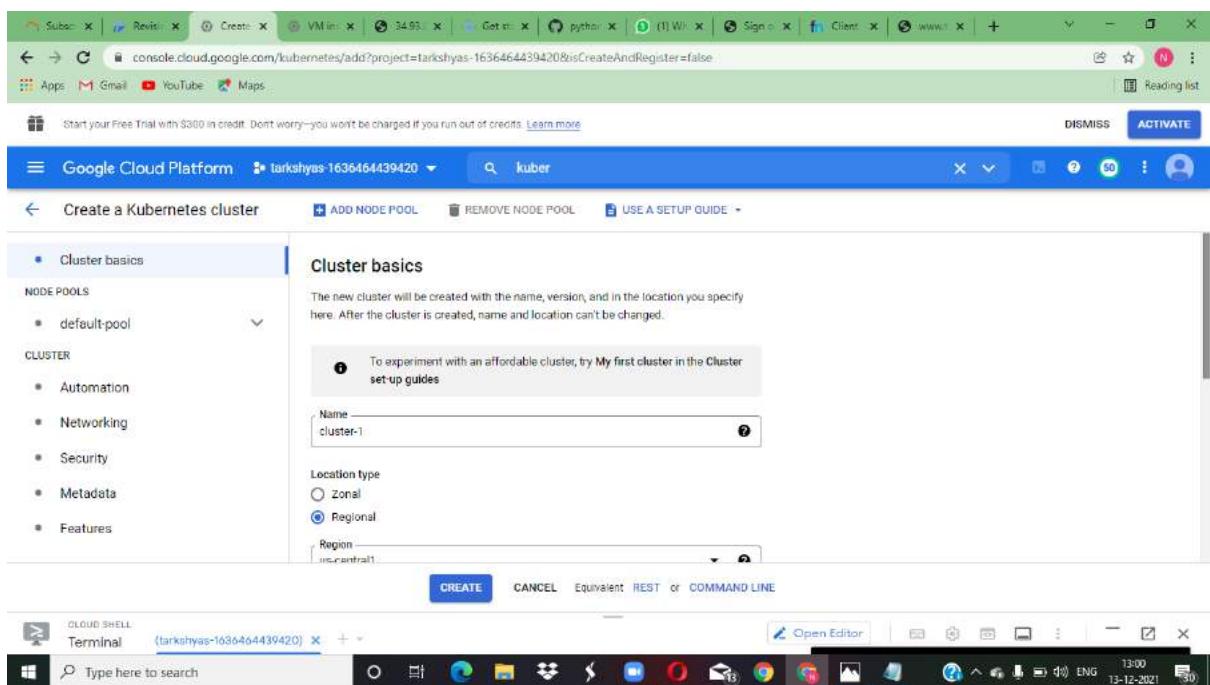
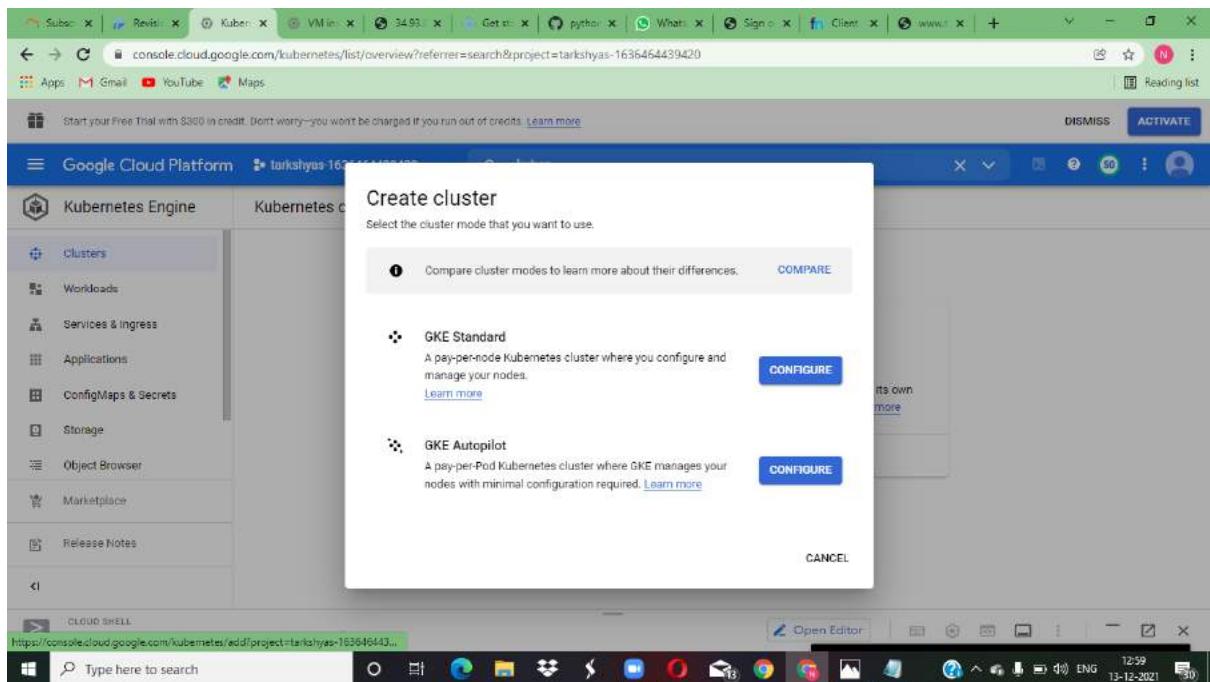
a9e482466809952c6e493441c507bad84c51aaef5ad532deh85c1led7c5a1b35
tscpc_user@38cloudshell:~ (tarkshyas-1636464439420)$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
a9e482466809 nginx:latest "/docker-entrypoint..." 19 seconds ago Up 18 seconds 0.0.0.0:8080->80/tcp unruffled_blackburn
tscpc_user@38cloudshell:~ (tarkshyas-1636464439420)$ vi index.html
tscpc_user@38cloudshell:~ (tarkshyas-1636464439420)$ vi index.html
tscpc_user@38cloudshell:~ (tarkshyas-1636464439420)$ docker cp index.html a9e482466809:/usr/share/nginx/html/
tscpc_user@38cloudshell:~ (tarkshyas-1636464439420)$ docker commit a9e482466809 cad/web:version1
sha256:d1b2c9016bbaf2f6092653d19eac2c479766cc7b9e1407e6feea00db7c536839
tscpc_user@38cloudshell:~ (tarkshyas-1636464439420)$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
cad/web version1 dib2c9016bb 25 seconds ago 141MB
nginx latest f652ca386ed1 10 days ago 141MB
tscpc_user@38cloudshell:~ (tarkshyas-1636464439420)$ docker tag cad/web:version1 us.gcr.io/tarkshyas-1636464439420/cad-site:version1
tscpc_user@38cloudshell:~ (tarkshyas-1636464439420)$ docker push us.gcr.io/tarkshyas-1636464439420/cad-site:version1
The push refers to repository [us.gcr.io/tarkshyas-1636464439420/cad-site]
d56431101bdb: Pushed
2bedd47a66c07: Layer already exists
82caad489ad7: Layer already exists
d3eldc44e82: Layer already exists
c9fd9c6c0d8: Layer already exists
0664b7821b60: Layer already exists
9321ff862abb: Layer already exists
version1: digest: sha256:90d350d088d85dfddbea665939ec6356ba3c7e01fb50a66858bb8c871aa44645 size: 1778
tscpc_user@38cloudshell:~ (tarkshyas-1636464439420)$

```

## Create Kubernetes Cluster by click on Kubernetes Cluster

The screenshot shows the Google Cloud Platform interface for creating a Kubernetes cluster. On the left, there's a sidebar with various options like Clusters, Workloads, Services & Ingress, Applications, ConfigMaps & Secrets, Storage, Object Browser, Marketplace, and Release Notes. The main area is titled "Kubernetes clusters" and contains a sub-section "Kubernetes Engine". It has three buttons at the bottom: "CREATE", "DEPLOY CONTAINER", and "TAKE THE QUICKSTART". Below these buttons, there's a note: "Containers package an application so it can easily be deployed to run in its own isolated environment. Containers are run on Kubernetes clusters." At the very bottom of the interface, there's a terminal window showing a command-line session.

## Create GKE Standard Cluster.



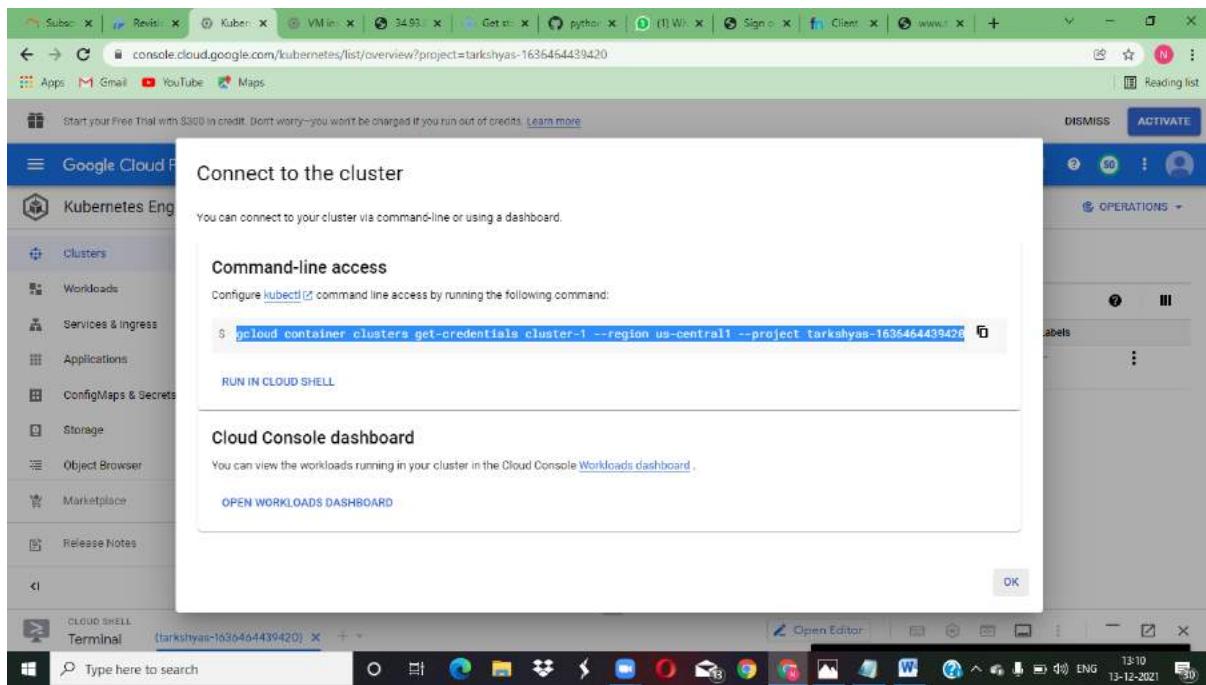
The screenshot shows the 'Create a Kubernetes cluster' page in the Google Cloud Platform. On the left, a sidebar lists 'Cluster basics' and sections for 'NODE POOLS' (default-pool), 'CLUSTER' (Automation, Networking, Security, Metadata, Features), and 'Control plane version'. Under 'Control plane version', it says 'Choose a release channel for automatic management of your cluster's version and upgrade cadence. Choose a static version for more direct management of your cluster's version.' It includes options for 'Static version' (radio button) and 'Release channel' (radio button selected), with 'Regular channel (default)' chosen. Below that is a dropdown for 'Version' set to '1.21.5-gke.1302 (default)'. A note at the bottom states: 'These versions have passed internal validation and are considered production-quality, but don't have enough historical data to guarantee their stability. Known issues generally have known workarounds. Release notes'.

At the bottom are 'CREATE', 'CANCEL', 'Equivalent REST or COMMAND LINE' buttons, and a 'Cloud Shell Terminal' window showing '(tarkshyas-1636464439420)'.

The screenshot shows the 'Kubernetes Engine' section of the Google Cloud Platform. On the left, a sidebar lists 'Clusters', 'Workloads', 'Services & Ingress', 'Applications', 'ConfigMaps & Secrets', 'Storage', 'Object Browser', 'Marketplace', and 'Release Notes'. The main area is titled 'Kubernetes clusters' with tabs for 'OVERVIEW', 'COST OPTIMIZATION', and 'PREVIEW'. It shows a table with one row for 'cluster-1':

Status	Name	Location	Number of nodes	Total vCPUs	Total memory	Notifications	Labels
Green checkmark	cluster-1	us-central1	9	18	36 GB	Low resource requests	-

Below the table, it says 'Last refreshed: 5 minutes ago'. At the bottom are 'Open Editor', 'Cloud Shell Terminal' window showing '(tarkshyas-1636464439420)', and a taskbar with various icons.



Create and expose the Kubernetes deployment for your docker image using following commands:

Use the kubectl expose command to generate a kubernetes service for the web-server deployment.

Use kubectl get pods command to see the pods created.

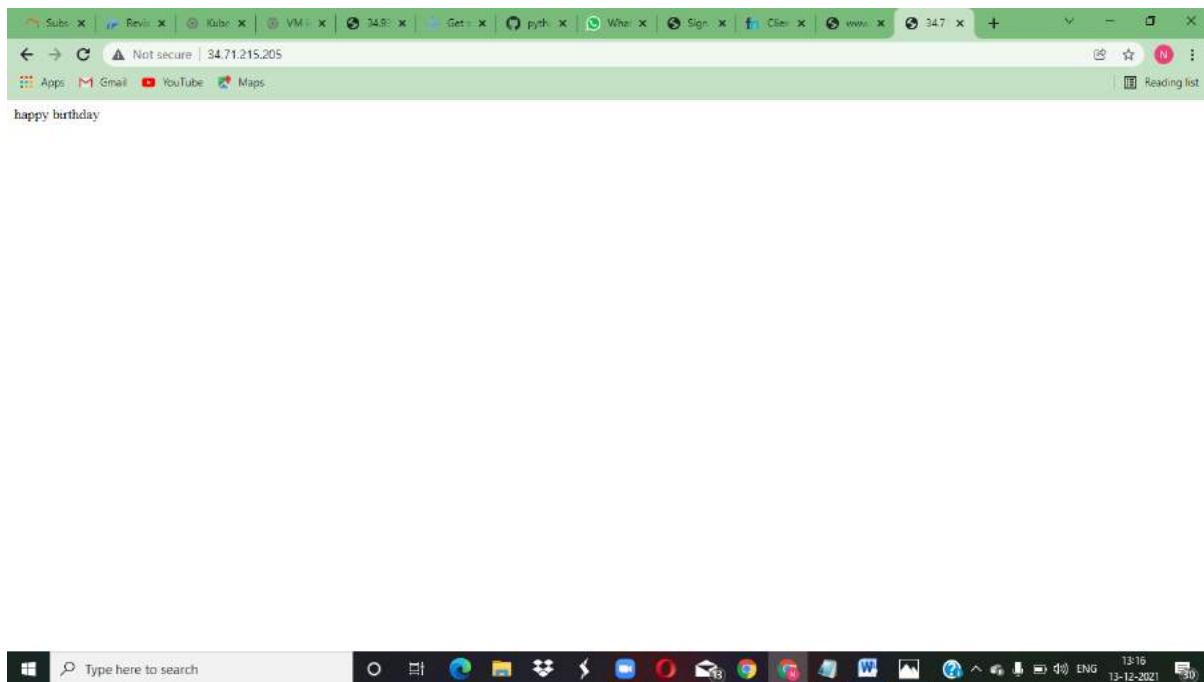
Use kubectl get service command to get the service.

Copy the External\_ip address to the clipboard to see the web application

The terminal window shows the following command sequence:

```
0664b7821b60: Layer already exists
9321ff862abb: Layer already exists
version1: digest: sha256:90d350d08d85df4dbca665939ec6356ba3c7e01fb50a66858bb8c871aa44645 size: 1778
tsgcp_user03@cloudshell:~ (tarkshyas-1636464439420)$ gcloud container clusters get-credentials cluster-1 --region us-central1 --project tarkshyas-1636464439420
Fetching cluster endpoint and auth data.
kubeconfig entry generated for cluster-1.
tsgcp_user03@cloudshell:~ (tarkshyas-1636464439420)$ kubectl create deployment web-server --image=us.gcr.io/tarkshyas-1636464439420/cad-site:v1
deployment.apps/web-server created
tsgcp_user03@cloudshell:~ (tarkshyas-1636464439420)$ kubectl expose deployment web-server --type LoadBalancer --port 80 --target-port 80
service/web-server exposed
tsgcp_user03@cloudshell:~ (tarkshyas-1636464439420)$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
web-server-574c49c59b-btgrf  1/1     Running   0          41s
tsgcp_user03@cloudshell:~ (tarkshyas-1636464439420)$ kubectl get service web-server
NAME         TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
web-server   LoadBalancer   10.64.7.107   <pending>   80:32048/TCP   39s
tsgcp_user03@cloudshell:~ (tarkshyas-1636464439420)$ kubectl get service web-server
NAME         TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
web-server   LoadBalancer   10.64.7.107   <pending>   80:32048/TCP   46s
tsgcp_user03@cloudshell:~ (tarkshyas-1636464439420)$ kubectl get service web-server
NAME         TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
web-server   LoadBalancer   10.64.7.107   34.71.215.205  80:32048/TCP   55s
tsgcp_user03@cloudshell:~ (tarkshyas-1636464439420)$
```

The terminal window is part of the Google Cloud Platform interface, with the title bar showing "Google Cloud Platform" and "tarkshyas-1636464439420". The bottom of the screen shows a Windows taskbar.



## 11. Using Terraform, Ansible create GCP resources (VM Instance, SQL Instance)

## Terraform

Create a new service account , Click on service account ->create service account.

Service accounts for project "tarkshyas-1636464439420"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. [Learn more about service accounts](#).

Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. [Learn more about service account organization policies](#).

Filter	Enter property name or value							
	Email	Status	Name	Description	Key ID	Key creation date	OAuth 2 Client ID	Actions
<input type="checkbox"/>	tarkshyas-1636464439420@appspot.gserviceaccount.com	✓	App Engine default service account	No keys			1090113967150	
<input type="checkbox"/>	24265042199-compute@developer.gserviceaccount.com	✓	Compute Engine default service account	No keys			1147114373514	

Subs x Revie x Create x VM x 24.80 x Get x pyth x What x Sign x f Client x www x 34.7 x + DISMISS ACTIVATE

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

Google Cloud Platform tarkshyas-1636464439420 service account

IAM & Admin Create service account

IAM Identity & Organization Policy Troubleshooter Policy Analyzer Organization Policies Service Accounts Workload Identity Federat... Labels Manage Resources Release Notes

Service account details

Service account name \* terra-account

Display name for this service account

Service acc... terra-account @tarkshyas-1636464439420.iam.gserviceaccount.co X C

Service account description

Describe what this service account will do

CREATE AND CONTINUE

Grant this service account access to project (optional)

Creating a service account for the project tarkshyas-1636464439420

Type here to search

13:28 13-12-2021

This screenshot shows the 'Create service account' page in the Google Cloud Platform IAM & Admin section. The 'Service account name' field is filled with 'terra-account'. The 'Service account description' field contains the placeholder 'Describe what this service account will do'. A 'CREATE AND CONTINUE' button is visible. Below this, there's a section for granting access to the project, which is currently empty.

Subs x Revie x Create x VM x 24.80 x Get x pyth x What x Sign x f Client x www x 34.7 x + DISMISS ACTIVATE

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

Google Cloud Platform tarkshyas-1636464439420 service account

IAM & Admin Create service account

IAM Identity & Organization Policy Troubleshooter Policy Analyzer Organization Policies Service Accounts Workload Identity Federat... Labels Manage Resources Release Notes

Service account details

Grant this service account access to project (optional)

Grant this service account access to tarkshyas-1636464439420 so that it has permission to complete specific actions on the resources in your project. [Learn more](#)

Role Owner Condition Add condition

Full access to all resources

+ ADD ANOTHER ROLE

CONTINUE

Grant users access to this service account (optional)

Creating a service account for the project tarkshyas-1636464439420

Type here to search

13:29 13-12-2021

This screenshot shows the continuation of the service account creation process. The 'Role' dropdown is set to 'Owner'. The 'Condition' section is present but empty. A 'CONTINUE' button is visible. Below this, there's a section for granting user access to the service account, which is currently empty.

## Create new JSON key.

The screenshot shows the Google Cloud Platform interface for managing service account keys. The left sidebar is titled 'IAM & Admin' and includes options like IAM, Identity & Organization, Policy Troubleshooter, Policy Analyzer, Organization Policies, Service Accounts (which is selected), Workload Identity Federation, Labels, Manage Resources, and Release Notes. The main content area is titled 'terra-account' and shows the 'Keys' tab selected. A warning message states: 'Service account keys could pose a security risk if compromised. We recommend you avoid downloading service account keys and instead use the Workload Identity Federation. You can learn more about the best way to authenticate service accounts on Google Cloud [here](#).'. Below this, there's a note: 'Add a new key pair or upload a public key certificate from an existing key pair.' and 'Block service account key creation using organization policies. Learn more about setting organization policies for service accounts.' There are two buttons: 'ADD KEY' (with dropdown) and 'Create new key' (selected). Other tabs include 'PERMISSIONS', 'DETAILS', 'METRICS', and 'LOGS'. At the bottom right of the main window, there are 'DISMISS' and 'ACTIVATE' buttons. The taskbar at the bottom shows various open windows and system icons.

This screenshot shows a modal dialog box titled 'Create private key for "terra-account"'. The dialog contains instructions: 'Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.' It includes a 'Key type' section with two options: 'JSON' (selected) and 'P12'. Below 'JSON' is the note: 'Recommended'. Below 'P12' is the note: 'For backward compatibility with code using the P12 format.' At the bottom of the dialog are 'CANCEL' and 'CREATE' buttons. The background of the main interface is visible, showing the same 'Service Accounts' screen as the previous screenshot.

## Using Terraform to create a VM in Google Cloud

Open Visual Studio code and create a new vm.tf file for the terraform config to create vm instance. The contents of this file describe all of the Google Cloud resources that will be used in the project.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named "TERRA" containing ".terraform", "spanner", "key.json", "terraform.exe", "vm.tf", and "vmprovider.tf".
- Editor:** The active tab is "vmprovider.tf", which contains the following code:

```
1 provider "google" {
2   credentials = file("key.json")
3   project     = "tankshyas-1636464430420"
4   region      = "us-central1"
5   zone        = "us-central1-c"
6 }
```
- Bottom Status Bar:** Shows "Ln 3, Col 37" and "Terraform".

Create provider file vmprovider.tf for creating vm instance.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named "TERRA" containing ".terraform", "spanner", "key.json", "terraform.exe", "vm.tf", and "vmprovider.tf".
- Editor:** The active tab is "vmprovider.tf", which contains the following code:

```
1 resource "google_compute_instance" "vm_instance" {
2   name            = "terraform-instance1"
3   machine_type   = "f1-micro"
4
5   boot_disk {
6     initialize_params {
7       image = "debian-cloud/debian-9"
8     }
9   }
10
11   network_interface {
12     # A default network is created for all GCP projects
13     network = "default"
14     access_config {
15     }
16   }
17 }
18 resource "google_compute_network" "vpc_network" {
19   name            = "terraform-network"
20   auto_create_subnetworks = "true"
21 }
```
- Bottom Status Bar:** Shows "Ln 21, Col 2" and "Terraform".

Open cloud SDK Shell and enter the following commands:

**Terraform init:** The terraform init command is used to initialize a working directory **containing** Terraform configuration files. This is the first command that should be run after writing a new Terraform configuration or cloning an existing one from version control.

**Terraform plan:** The terraform plan command creates an execution plan, which lets you preview the changes that Terraform plans to make to your infrastructure.

Terraform apply: The terraform apply command executes the actions proposed in a Terraform plan.

```
Google Cloud SDK Shell
C:\Users\user\AppData\Local\Google\Cloud SDK>cd ..
C:\Users\user\AppData\Local\Google>cd ..
C:\Users\user\AppData\local>cd ..
C:\Users\user\AppData>cd ..
C:\Users\user>clear
'clear' is not recognized as an internal or external command,
operable program or batch file.
C:\Users\user>cd Desktop
C:\Users\user\Desktop>cd terra
C:\Users\user\Desktop\terra>terraform init
Initializing the backend...
Initializing provider plugins...
The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.google: version = ">= 3.65"

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
run this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

C:\Users\user\Desktop\terra>
```

```
Google Cloud SDK Shell
    + node_affinities {
        + key      = (known after apply)
        + operator = (known after apply)
        + values   = (known after apply)
    }
}
# google_compute_network.vpc_network will be created
resource "google_compute_network" "vpc_network" {
    + auto_create_subnetworks      = true
    + delete_default_routes_on_create = false
    + gateway_ipv4                = (known after apply)
    + id                           = (known after apply)
    + mtu                          = (known after apply)
    + name                         = "terraform-network"
    + project                      = (known after apply)
    + routing_mode                 = (known after apply)
    + self_link                     = (known after apply)
}

Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

Enter a value: yes

google_compute_instance.vm_instance: Creating...
google_compute_network.vpc_network: Creating...
google_compute_network.vpc_network: Still creating... [10s elapsed]
google_compute_instance.vm_instance: Still creating... [10s elapsed]
google_compute_instance.vm_instance: Still creating... [20s elapsed]
google_compute_network.vpc_network: Still creating... [20s elapsed]
google_compute_network.vpc_network: Still creating... [30s elapsed]
google_compute_instance.vm_instance: Still creating... [30s elapsed]
google_compute_instance.vm_instance: Creation complete after 39s [id=projects/tarkshyas-1636464439420/zones/us-central1-c/instances/terraform-instance1]
google_compute_network.vpc_network: Still creating... [40s elapsed]
google_compute_network.vpc_network: Creation complete after 45s [id=projects/tarkshyas-1636464439420/global/networks/terraform-network]

apply complete! Resources: 2 added, 0 changed, 0 destroyed.

C:\Users\user\Desktop\terra>
```

The screenshot shows the Google Cloud Platform Compute Engine VM instances page. On the left, a sidebar lists options like VM instances, Instance templates, Sole-tenant nodes, Machine images, TPUs, Committed use discounts, Migrate for Compute Engi., Marketplace, and Release Notes. The main area displays a table of VM instances with columns for Status, Name, and Zone. One instance, 'terraform-instance1', is listed under the 'us-central1-c' zone. A modal window titled 'Select an instance' is open, prompting the user to 'Please select at least one resource.' To the right, there's a sidebar with 'Start your project' instructions, links to 'Connect to your instances' and 'Transfer files', and sections for 'Find existing VM solutions', 'Explore Marketplace', 'Quickly learn how to build a two-tier web app', 'How-to guides and tutorials', and 'Explore tools and APIs'.

Likewise SQL instance is created in terraform by using provider and configuration files.

The screenshot shows a Visual Studio Code editor window with several tabs open. The tabs include 'sql.tf - terra - Visual Studio Code', 'vm.tf', 'vmprovider.tf', 'sqlprovider.tf', and 'key.json'. The 'sql.tf' tab contains Terraform code for creating a SQL instance:

```
provider "google" {
  credentials = file("key.json")
  project    = "tarkshyas-tarkshyas-1636464439420"
  region     = "us-central1"
  zone       = "us-central1-c"
}
```

The left side of the screen shows the VS Code interface with the Explorer, Outline, and Search panes visible. The status bar at the bottom indicates the file is 'sql.tf' and the encoding is 'UTF-8'.

A screenshot of Visual Studio Code showing a file tree and several code editors. The file tree on the left shows a directory structure under 'TERRA' containing '.terraform', 'spanner', 'sql', and 'sqlprovider'. The 'sqlprovider' folder contains files like 'key.json', 'sql.tf', 'sqlprovider.tf', 'spf.t', 'spprovider.tf', 'terrafrom.exe', 'terrafrom.state', 'vm.tf', and 'vmprovider.tf'. One editor tab is open with the file 'sqlprovider.tf', displaying Terraform code for creating a database, private network, and VPC peering. Another editor tab shows 'vm.tf'. The status bar at the bottom indicates 'In 1, Col 1' with 'Spaces: 2', 'UTF-8', 'CRLF', 'Terraform', and the date '13-12-2021'.

```
resource "google_sql_database" "database" {
  name = "my-database"
  instance = google_sql_database_instance.instance.name
}
resource "google_compute_network" "private_network" {
  provider = google-beta
  name = "private-network"
}
resource "google_compute_global_address" "private_ip_address" {
  provider = google-beta
  name = "private-ip-address"
  purpose = "VPC_PEERING"
  address_type = "INTERNAL"
  prefix_length = 16
  network = google_compute_network.private_network.id
}
resource "google_serviceNetworking_connection" "private_vpc_connection" {
  provider = google-beta
  network = google_compute_network.private_network.id
  service = "servicenetworking.googleapis.com"
  reserved_peering_ranges = [google_compute_global_address.private_ip_address.name]
}
resource "random_id" "db_name_suffix" {
  byte_length = 4
}
```

A screenshot of the Google Cloud SDK Shell window showing the output of the 'terraform init' command. It displays the creation progress of various Google Compute Engine resources like VM instances and VPC networks. It also shows the initialization of the backend and provider plugins, and provides instructions for beginning work with Terraform. The status bar at the bottom indicates 'In 1, Col 1' with 'Spaces: 2', 'UTF-8', 'CRLF', 'Terraform', and the date '13-12-2021'.

```
Google Cloud SDK Shell
google_compute_instance.vm_instance: Still creating... [10s elapsed]
google_compute_instance.vm_instance: Still creating... [20s elapsed]
google_compute_network.vpc_network: Still creating... [20s elapsed]
google_compute_network.vpc_network: Still creating... [30s elapsed]
google_compute_instance.vm_instance: Still creating... [30s elapsed]
google_compute_instance.vm_instance: Creation complete after 39s [id:projects/tarkshyas-1636464439420/zones/us-central1-c/instances/terraform-instance]
google_compute_network.vpc_network: Still creating... [40s elapsed]
google_compute_network.vpc_network: Creation complete after 45s [id:projects/tarkshyas-1636464439420/global/networks/terraform-network]

apply completed: Resources: 2 added, 0 changed, 0 destroyed.

C:\Users\user\Desktop\terra>cd spanner
C:\Users\user\Desktop\terra\spanner>cd sql
C:\Users\user\Desktop\terra\spanner\sql>terraform init
Initializing the backend...
Initializing provider plugins...
The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "... constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.google: version = "> 3.65"
* provider.google-beta: version = "> 3.65"
* provider.random: version = "> 3.1"

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

C:\Users\user\Desktop\terra\spanner\sql>
```

The screenshot shows the Google Cloud Platform SQL Instances page. It lists two instances:

Instance ID	Type	Public IP address	Private IP address	Instance connection name
private-instance-e54b2a43	MySQL 5.7	10.23.0.8		tarkshyas-1636464439420us-central1:private-instance-e54b2a43
temp	MySQL 8.0	35.224.28.120		tarkshyas-16364644...

Below the table, a message says "No instances selected". There are tabs for MONITORING and LABELS.

## Ansible

Install ansible version 2.9.0. And install necessary configurations as follows:

The screenshot shows a Google Cloud Shell terminal window. The user runs the command `pip install ansible==2.9.0`. The output shows that Python 2 is deprecated and users should upgrade to Python 3. It also lists various package dependencies being installed.

```
tsgcp_user03@cloudshell:~ (tarkshyas-1636464439420)$ pip install ansible==2.9.0
*****
Python 2 is deprecated. Upgrade to pip3 as soon as possible.
See https://cloud.google.com/python/docs/python2-sunset

To suppress this warning, create an empty ~/.cloudshell/no-pip-warning file.
The command will automatically proceed in 5 seconds or on any key.
*****
Requirement already satisfied: ansible==2.9.0 in ./local/lib/python2.7/site-packages (2.9.0)
Requirement already satisfied: cryptography in ./local/lib/python2.7/site-packages (from ansible==2.9.0) (3.3.2)
Requirement already satisfied: PyYAML in ./local/lib/python2.7/site-packages (from ansible==2.9.0) (5.4.1)
Requirement already satisfied: jinja2 in ./local/lib/python2.7/site-packages (from ansible==2.9.0) (2.11.3)
Requirement already satisfied: six>=1.4.1 in /usr/local/lib/python2.7/dist-packages (from cryptography->ansible==2.9.0) (1.16.0)
Requirement already satisfied: enum34; python_version < "3" in /usr/local/lib/python2.7/dist-packages (from cryptography->ansible==2.9.0) (1.1.10)
Requirement already satisfied: ipaddress; python_version < "3" in ./local/lib/python2.7/site-packages (from cryptography->ansible==2.9.0) (1.0.23)
Requirement already satisfied: cffi>=1.12 in ./local/lib/python2.7/site-packages (from cryptography->ansible==2.9.0) (1.15.0)
Requirement already satisfied: MarkupSafe>=0.23 in ./local/lib/python2.7/site-packages (from jinja2->ansible==2.9.0) (1.1.1)
Requirement already satisfied: pyparser in ./local/lib/python2.7/site-packages (from cffi>=1.12->cryptography->ansible==2.9.0) (2.21)
tsgcp_user03@cloudshell:~ (tarkshyas-1636464439420)$
```

```
1.1.10)
Requirement already satisfied: ipaddress; python_version < "3" in ./local/lib/python2.7/site-packages (from cryptography->ansible==2.9.0)
(1.0.23)
Requirement already satisfied: cffi!=1.12 in ./local/lib/python2.7/site-packages (from cryptography->ansible==2.9.0) (1.15.0)
Requirement already satisfied: MarkupSafe>=0.23 in ./local/lib/python2.7/site-packages (from jinja2->ansible==2.9.0) (1.1.1)
Requirement already satisfied: pycparser in ./local/lib/python2.7/site-packages (from cffi!=1.12->cryptography->ansible==2.9.0) (2.21)
tsgcp_user03@cloudshell:~ (tarkshyas-1636464439420)$ ansible --version
/home/tsgcp_user03/.local/lib/python2.7/site-packages/ansible/parsing/vault/_init_.py:41: CryptographyDeprecationWarning: Python 2 is no
longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
    from cryptography.exceptions import InvalidSignature
ansible 2.9.0
  config file = None
  configured module search path = [u'/home/tsgcp_user03/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /home/tsgcp_user03/.local/lib/python2.7/site-packages/ansible
  executable location = /home/tsgcp_user03/.local/bin/ansible
  python version = 2.7.16 (default, Oct 10 2019, 22:02:15) [GCC 8.3.0]
tsgcp_user03@cloudshell:~ (tarkshyas-1636464439420)$ pip install requests google-auth
*****
Python 2 is deprecated. Upgrade to pip3 as soon as possible.
See https://cloud.google.com/python/docs/python2-sunset

To suppress this warning, create an empty ~/.cloudshell/no-pip-warning file.
The command will automatically proceed in 5 seconds or on any key.
*****
```

```
1.1.10)
Requirement already satisfied: certifi==2017.4.17 in /usr/local/lib/python2.7/dist-packages (from requests) (2021.10.8)
Requirement already satisfied: idna<3,>=2.5; python_version < "3" in /usr/local/lib/python2.7/dist-packages (from requests) (2.10)
Requirement already satisfied: chardet<5,>=3.0.2; python_version < "3" in /usr/local/lib/python2.7/dist-packages (from requests) (4.0.0)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python2.7/dist-packages (from google-auth) (1.16.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python2.7/dist-packages (from google-auth) (0.2.8)
Requirement already satisfied: enum34>=1.1.10; python_version < "3.4" in /usr/local/lib/python2.7/dist-packages (from google-auth) (1.1.10)
Requirement already satisfied: setuptools>=40.3.0 in /usr/local/lib/python2.7/dist-packages (from google-auth) (44.1.1)
Requirement already satisfied: rsa<4.6; python_version < "3.6" in /usr/local/lib/python2.7/dist-packages (from google-auth) (4.5)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/python2.7/dist-packages (from google-auth) (3.1.1)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python2.7/dist-packages (from pyasn1-modules>=0.2.1->google-auth) (0.
4.8)
tsgcp_user03@cloudshell:~ (tarkshyas-1636464439420)$ ansible-galaxy collection install community.mysql
/home/tsgcp_user03/.local/lib/python2.7/site-packages/ansible/parsing/vault/_init_.py:41: CryptographyDeprecationWarning: Python 2 is no
longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
    from cryptography.exceptions import InvalidSignature
Process install dependency map
Process install dependency map
Starting collection install process
Installing 'community.mysql:3.0.0' to '/home/tsgcp_user03/.ansible/collections/ansible_collections/community/mysql'
tsgcp_user03@cloudshell:~ (tarkshyas-1636464439420)$
```

```

Requirement already satisfied: setuptools>=40.3.0 in /usr/local/lib/python2.7/dist-packages (from google-auth) (44.1.1)
Requirement already satisfied: rsa<4.6; python_version < "3.6" in /usr/local/lib/python2.7/dist-packages (from google-auth) (4.5)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/python2.7/dist-packages (from google-auth) (3.1.1)
Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in /usr/local/lib/python2.7/dist-packages (from pyasn1-modules>=0.2.1>google-auth) (0.4.6)
tagcp user03@cloudshell:~ (tarkshyas-1636464439420)$ ansible-galaxy collection install community.mysql
/home/tsgcp_user03/.local/lib/python2.7/site-packages/ansible/parsing/vault/_init__.py:41: CryptographyDeprecationWarning: Python 2 is no
longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
  from cryptography.exceptions import InvalidSignature
Process install dependency map
tagcp user03@cloudshell:~ (tarkshyas-1636464439420)$ ansible-galaxy collection install community.mysql
/home/tsgcp_user03/.local/lib/python2.7/site-packages/ansible/parsing/vault/_init__.py:41: CryptographyDeprecationWarning: Python 2 is no
longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
  from cryptography.exceptions import InvalidSignature
Process install dependency map
Starting collection install process
Installing 'community.mysql:3.0.0' to '/home/tsgcp_user03/.ansible/collections/ansible_collections/community/mysql'
tagcp user03@cloudshell:~ (tarkshyas-1636464439420)$ mkdir ansible
mkdir: cannot create directory 'ansible': File exists
tagcp user03@cloudshell:~ (tarkshyas-1636464439420)$ mkdir ansi
tagcp user03@cloudshell:~ (tarkshyas-1636464439420)$ cd ansi
tagcp user03@cloudshell:~/ansi (tarkshyas-1636464439420)$ ls
tagcp user03@cloudshell:~/ansi (tarkshyas-1636464439420)$ cd ..
tagcp user03@cloudshell:~ (tarkshyas-1636464439420)$ []

```

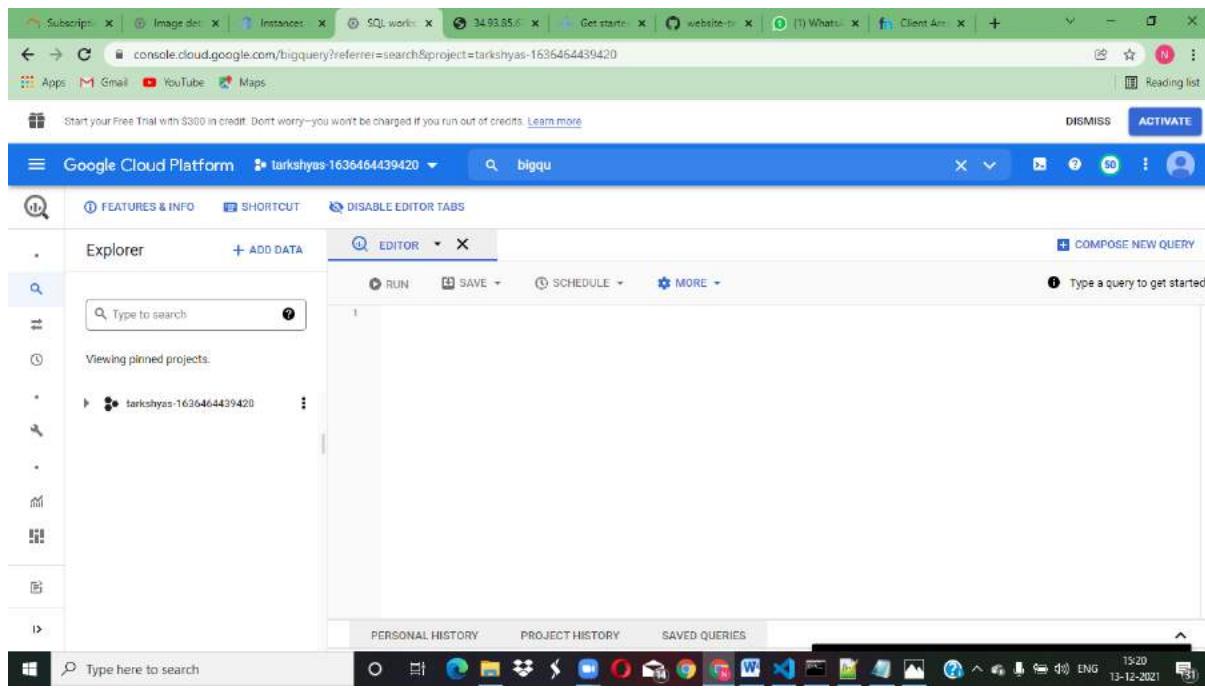
The screenshot shows a terminal session within the Google Cloud Platform interface. The user is installing the 'community.mysql' collection. The terminal output indicates that the collection is being installed into the 'ansi' directory. A file transfer window is also visible, showing two files transferred: 'key.json' and 'sqlnew.yml'. The system tray at the bottom right shows the date and time as 13-12-2021 14:18.

Thus instance temp-3 is created.

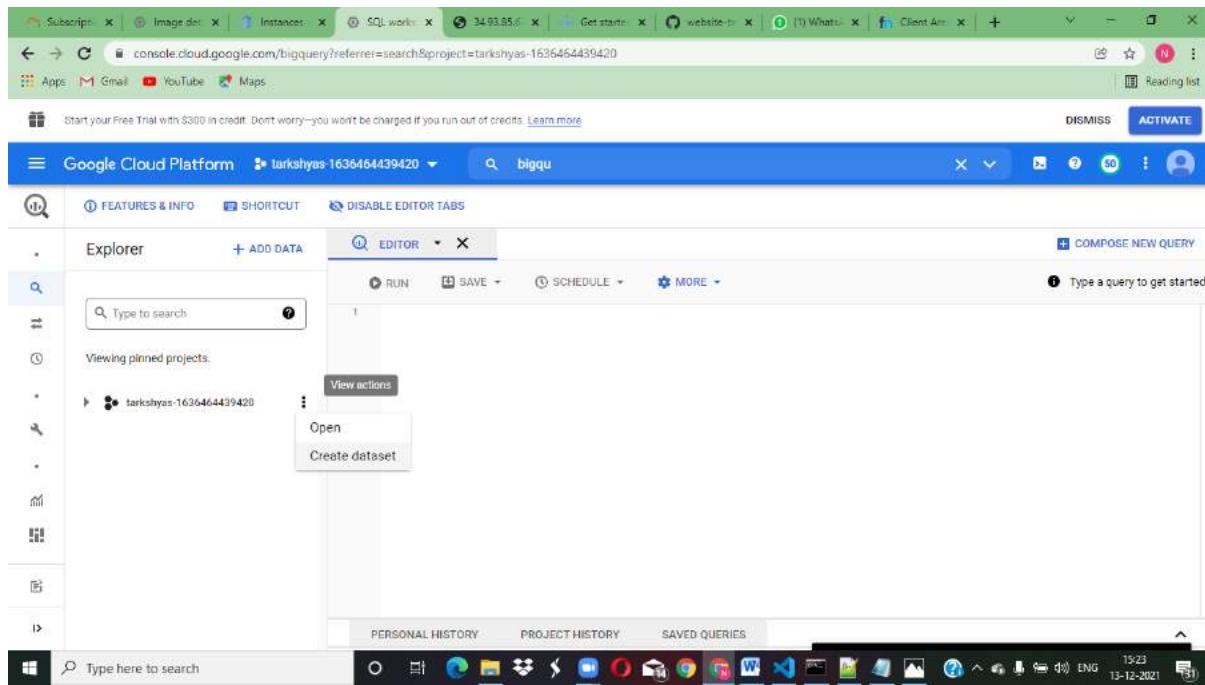
Instance ID	Type	Public IP address	Private IP address	Instance connection name	Actions
private-instance-e54b2a43	MySQL 5.7	10.23.0.3	tarkshyas-16364644...	<input checked="" type="checkbox"/>	<span>⋮</span>
temp	MySQL 8.0	35.224.28.120	tarkshyas-16364644...	<input checked="" type="checkbox"/>	<span>⋮</span>
temp-3	MySQL 5.7	34.121.146.196	tarkshyas-16364644...	<input checked="" type="checkbox"/>	<span>⋮</span>

The screenshot shows the 'Instances' tab in the Google Cloud Platform SQL section. It lists three instances: 'private-instance-e54b2a43', 'temp', and 'temp-3'. The 'temp-3' instance is highlighted with a green checkmark. The system tray at the bottom right shows the date and time as 13-12-2021 14:23.

## 12. Understanding the concept of BigQuery. Import your data into BigQuery and Cloud Storage; SQL basics(WHERE, GROUP BY, JOIN TABLES); SQL BigQuery specific features(Partitioning, Clustering); BigQuery ML Models

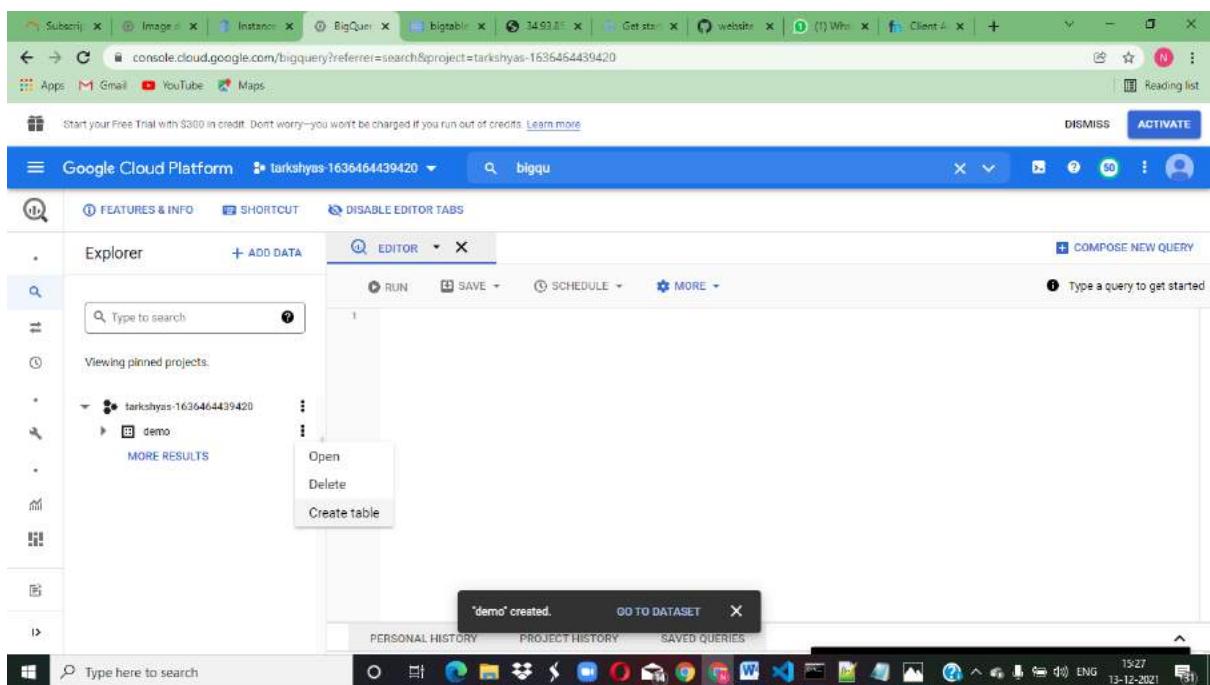
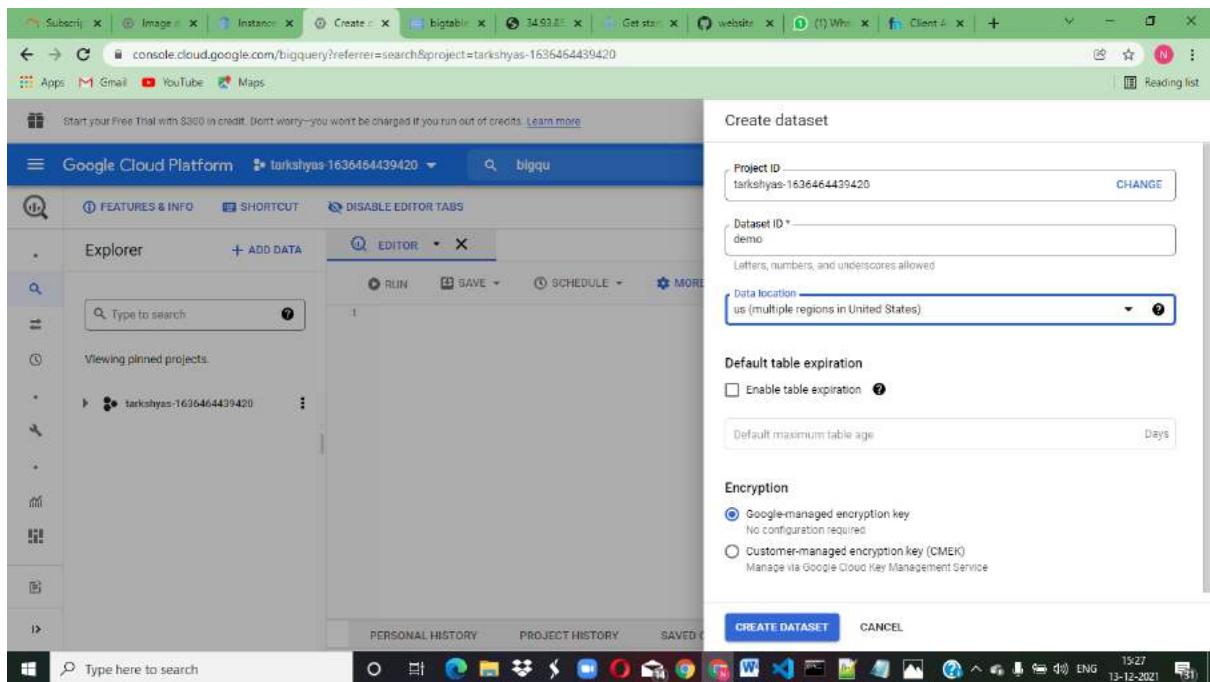


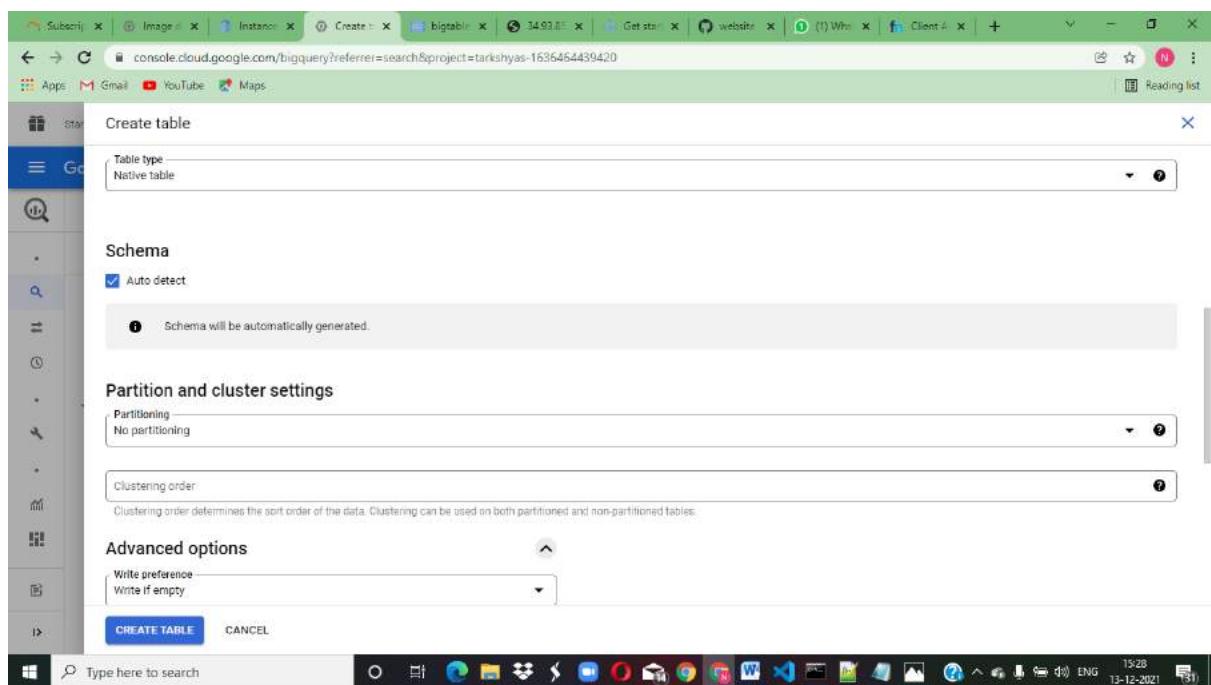
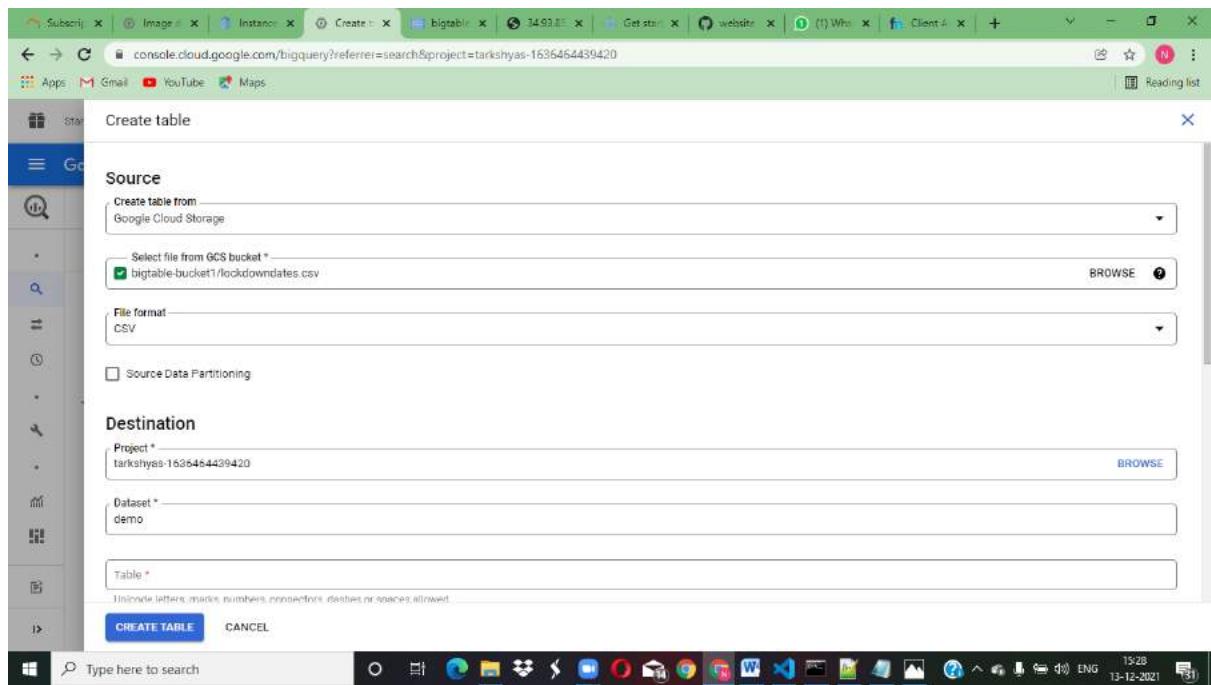
Create dataset

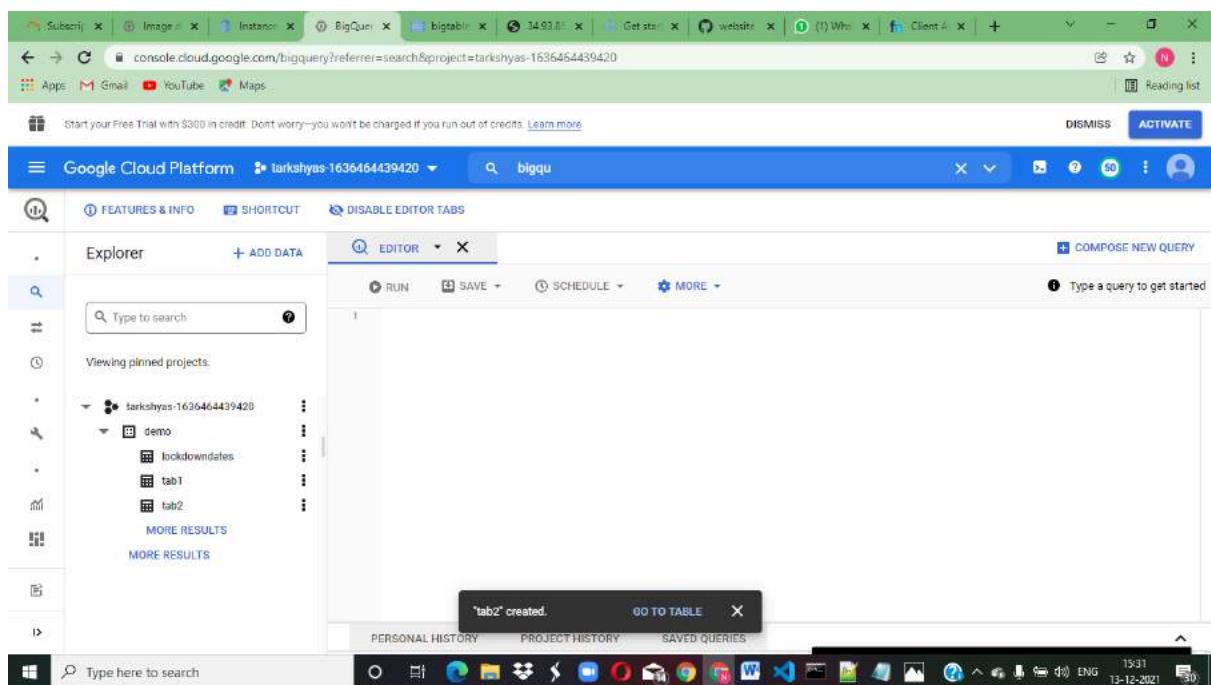
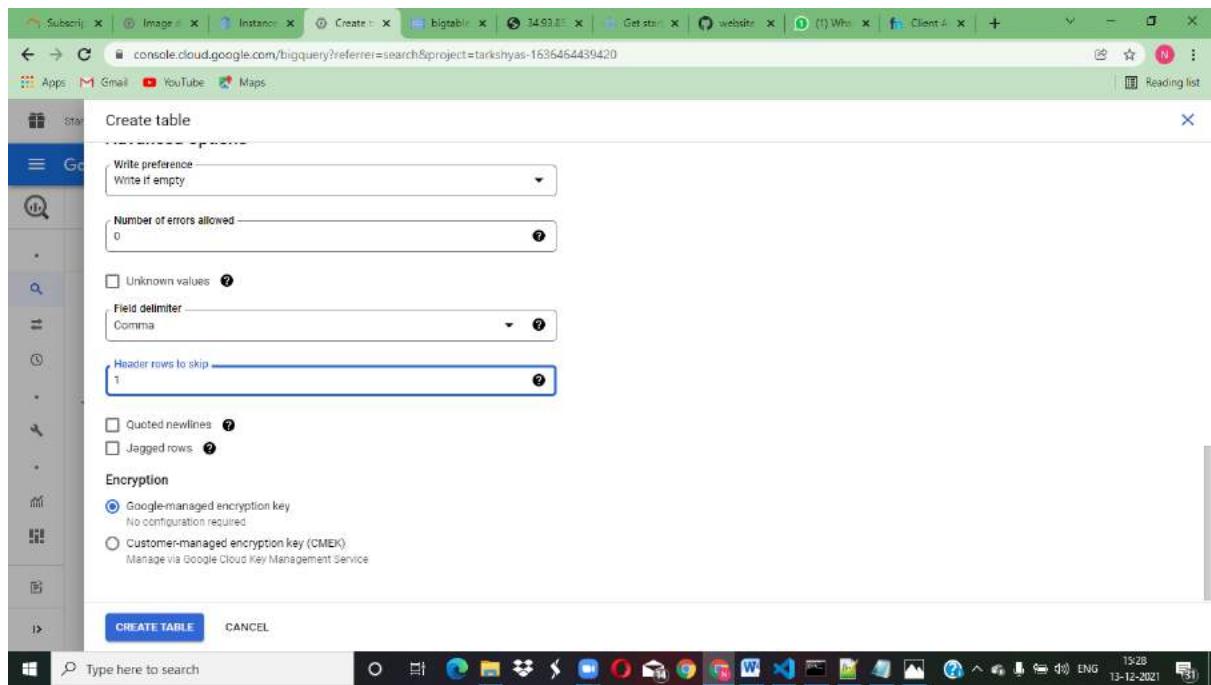


The screenshot shows the Google Cloud Platform Cloud Storage interface. On the left, a sidebar lists 'Cloud Storage', 'Browser', 'Monitoring', and 'Settings'. The main area displays 'Bucket details' for 'bigtable-bucket1'. The bucket's location is 'us (multiple regions in United States)', storage class is 'Standard', public access is 'Not public', and protection is 'None'. The 'OBJECTS' tab is selected, showing a list of objects: 'lockdowndates.csv' (15.9 KB, application/vnd.ms-excel), 'table1.txt' (45 B, text/plain), and 'table2.txt' (73 B, text/plain). A toolbar below the list includes 'UPLOAD FILES', 'UPLOAD FOLDER', 'CREATE FOLDER', 'MANAGE HOLDS', 'DOWNLOAD', and 'DELETE'. A 'Filter by name prefix only' dropdown and a 'Filter objects and folders' button are also present. The bottom of the screen shows a Windows taskbar with various icons.

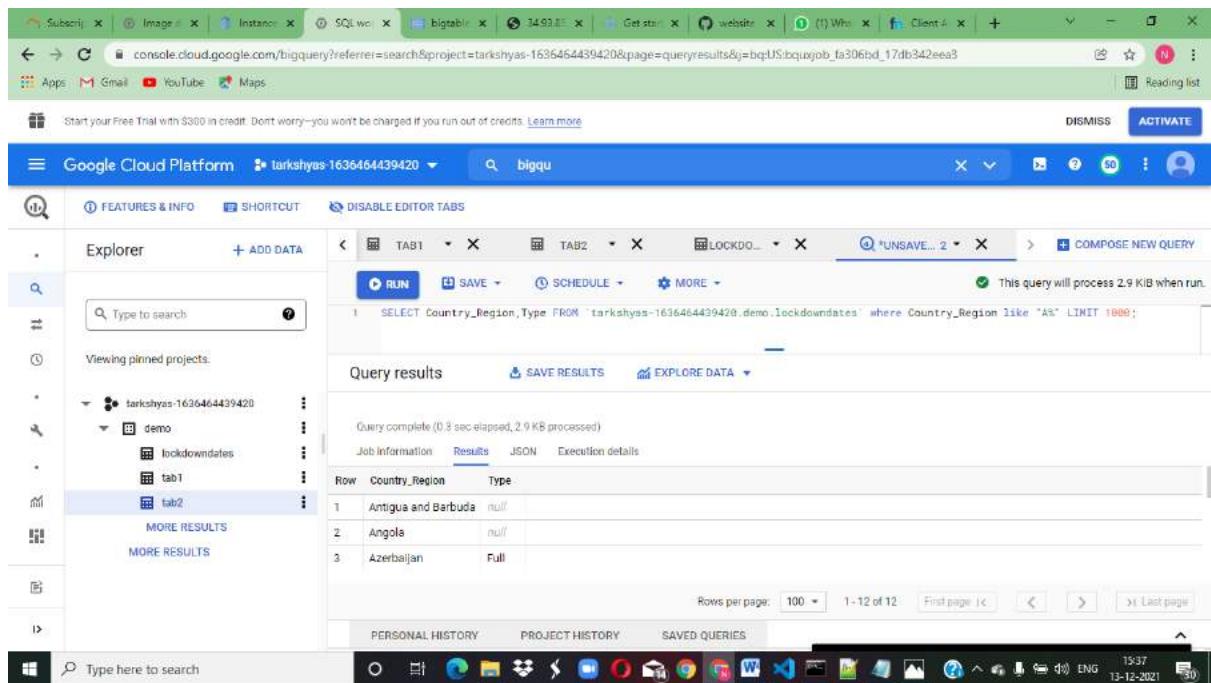
The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar includes 'FEATURES & INFO', 'SHORTCUT', 'DISABLE EDITOR TABS', 'Explorer', and 'ADD DATA'. The main area has tabs for 'EDITOR' (selected), 'RUN', 'SAVE', 'SCHEDULE', and 'MORE'. A search bar at the top right says 'Type a query to get started'. A dropdown menu is open over a project named 'tarkshyas-1636464439420', with options 'Open' and 'Create dataset'. At the bottom, there are tabs for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES'. The bottom of the screen shows a Windows taskbar with various icons.







## WHERE Clause



The screenshot shows the Google Cloud Platform BigQuery interface. On the left, the Explorer sidebar displays a project named 'tarkshyas-1636464439420' with a 'demo' dataset containing tables 'lockdowndates', 'tab1', and 'tab2'. The main query editor window contains the following SQL code:

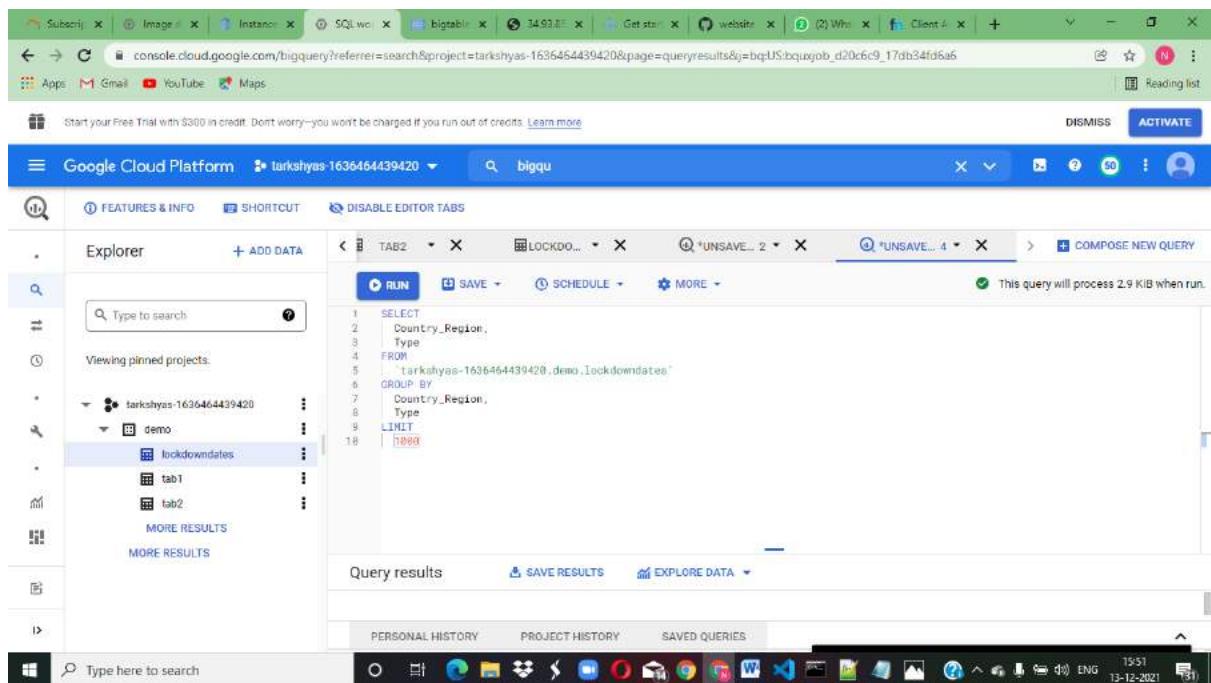
```
1 SELECT Country_Region, Type FROM `tarkshyas-1636464439420.demo.lockdowndates` WHERE Country_Region like "%A%" LIMIT 1000;
```

The results pane shows the output of the query:

Row	Country_Region	Type
1	Antigua and Barbuda	null
2	Angola	null
3	Azerbaijan	Full

At the bottom of the interface, the taskbar shows various open applications including Microsoft Word, Excel, and File Explorer.

## GROUP BY



The screenshot shows the Google Cloud Platform BigQuery interface. On the left, the Explorer sidebar displays a project named 'tarkshyas-1636464439420' with a 'demo' dataset containing tables 'lockdowndates', 'tab1', and 'tab2'. The main query editor window contains the following SQL code:

```
1 SELECT
2   Country_Region,
3   Type
4   FROM
5   `tarkshyas-1636464439420.demo.lockdowndates`
6   GROUP BY
7   Country_Region,
8   Type
9   LIMIT
10  | 1000|
```

The results pane shows the output of the query:

Country_Region	Type
Antigua and Barbuda	null
Angola	null
Azerbaijan	Full

At the bottom of the interface, the taskbar shows various open applications including Microsoft Word, Excel, and File Explorer.

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays a project named 'tarkshyas-1636464439420' with datasets 'demo', 'tab1', and 'tab2'. The main area shows a query editor with the following code:

```
1 SELECT
2   Country_Region,
3   Type
```

The results pane shows the following data:

Row	Country_Region	Type
1	Brazil	null
2	Sweden	null
3	Belarus	null

At the bottom, there is a toolbar with various icons and a system tray showing the date and time.

## JOIN

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays a project named 'tarkshyas-1636464439420' with datasets 'demo', 'tab1', and 'tab2'. The main area shows a query editor with the following code:

```
1 SELECT
2   a.name,
3   a.dept_id,
4   b.subject
5 FROM
6   [tarkshyas-1636464439420].demo.tab1 AS a
7   JOIN
8   [tarkshyas-1636464439420].demo.tab2 AS b
9   ON
10    a.stud_id=b.stud_id
11   LIMIT
12    1000
```

At the bottom, there is a toolbar with various icons and a system tray showing the date and time.

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with a pinned project 'tarkshyas-1636464439420' containing a 'demo' dataset with tables 'lockdowndates', 'tab1', and 'tab2'. The main area shows a query editor with the following SQL code:

```
1 SELECT
2   a.name,
3   b.dept_id,
```

The results of the query are displayed in a table:

Row	name	dept_id	subject
1	nandhu	111	science
2	paaru	112	commerce
3	renju	113	commerce

The status bar at the bottom indicates '16:14 13-12-2021'.

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with a pinned project 'tarkshyas-1636464439420' containing a 'demo' dataset with tables 'lockdowndates', 'tab1', and 'tab2'. The main area shows a query editor with the following SQL code:

```
1 CREATE TABLE
2   tarkshyas-1636464439420.demo.lockdowndates_cleaned AS
3   SELECT
4     Country_Region,
5     Type,
6     Reference,
7     DATE(Date) AS news_date
8   FROM
9     tarkshyas-1636464439420.demo.lockdowndates
```

The status bar at the bottom indicates '16:21 13-12-2021'.

The screenshot shows the Google Cloud Platform BigQuery interface. In the left sidebar, under the 'tarkshyas-1636464439420' project, the 'demo' dataset is selected. Inside 'demo', there are two tables: 'lockdowndates' and 'lockdowndates\_cleaned'. A new table, 'tab2', is currently being created. The main pane displays the SQL query:

```
CREATE TABLE
  tarkshyas-1636464439420.demo.lockdowndates_cleaned AS
SELECT
```

The status bar at the bottom right indicates '16:21 13-12-2021'.

The screenshot shows the Google Cloud Platform BigQuery interface after the table creation query has been run. The 'lockdowndates\_cleaned' table is now listed in the 'demo' dataset. The table has 86 rows and 4 columns: 'id', 'country', 'type', and 'url'. The first few rows of data are:

id	country	type	url
79	Peru	Full	<a href="https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Peru">https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Peru</a>
80	Vatican City	Full	<a href="https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Vatican_City">https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Vatican_City</a>
81	Mongolia	Full	<a href="https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Mongolia">https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Mongolia</a>
82	Bosnia and Herzegovina	Full	<a href="https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Bosnia_and_Herzegovina">https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Bosnia_and_Herzegovina</a>
83	Qatar	Full	<a href="https://www.theguardian.com/global-development/2020/mar/20/covid-19-lockdown-turns-qatars-largest-migrant-camp-into-a-quarantine-island">https://www.theguardian.com/global-development/2020/mar/20/covid-19-lockdown-turns-qatars-largest-migrant-camp-into-a-quarantine-island</a>
84	Slovenia	Full	<a href="https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Slovenia">https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Slovenia</a>
85	Denmark	Full	<a href="https://www.businessinsider.com/countries-on-lockdown-coronavirus-italy-2020-3?r=US&amp;lR=T#many-countries-have-also-imposed-lockdowns">https://www.businessinsider.com/countries-on-lockdown-coronavirus-italy-2020-3?r=US&amp;lR=T#many-countries-have-also-imposed-lockdowns</a>
86	Maldives	Full	<a href="https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_the_Maldives">https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_the_Maldives</a>

The status bar at the bottom right indicates '16:24 13-12-2021'.

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with a tree view of projects and datasets. A specific dataset 'tarkshyas-1636464439420' is expanded, showing a 'demo' dataset which contains a table named 'lockdowndates\_cleaned'. The main panel shows the schema of this table, which includes columns like 'ID643.html', 'virus\_pandemic\_in\_the\_State\_of\_Palestine', 'virus\_pandemic\_in\_Bhutan', 'virus\_pandemic\_in\_Albania', 'coronavirus\_lockdown', 'virus\_pandemic\_in\_Slovakia', 'virus\_pandemic\_in\_Peru', 'virus\_pandemic\_in\_Vatican\_City', and 'virus\_pandemic\_in\_Morocco'. The table was last modified on 2020-03-02.

## PARTITIONING OF TABLES

The screenshot shows the Google Cloud Platform BigQuery interface with a query editor open. The query is:

```
1 CREATE TABLE
2   tarkshyas-1636464439420.demo.lockdowndates_cleaned_partitioned partition by (news_date) AS
3 SELECT
4   *
5 FROM
6   tarkshyas-1636464439420.demo.lockdowndates_cleaned
```

The status bar indicates that the query will process 15.5 KB when run. The processing location is set to US. The main panel shows the 'Query results' section, which is currently empty.

The screenshot shows the Google Cloud Platform BigQuery interface. The top navigation bar includes tabs for Subscriptions, Images, Instances, SQL, bigtable, 34.99.0, Get started, website, WhatsApp, Client, and a Reading list. Below the navigation bar, a message encourages starting a free trial with \$300 in credit. The main header displays "Google Cloud Platform" and the project ID "tarkshyas-1636464439420". A search bar is present, followed by a "bigqu" button.

The left sidebar features an "Explorer" section with a search bar and a list of pinned projects. Under the current project, there is a "demo" folder containing tables like "lockdowndates", "lockdowndates\_cleaned", and "lockdowndates\_cleaned\_partitioned".

The right pane shows the details for the "lockdowndates\_cleaned\_partitioned" table. It is identified as a partitioned table. The "PREVIEW" tab is selected, displaying a list of 5 rows with columns "Row", "Country\_Region", "Type", and "Reference".

Row	Country_Region	Type	Reference
1	Barbados	Full	<a href="https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Barbados">https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Barbados</a>
2	Cambodia	Partial	<a href="https://www.bangkokpost.com/world/1888300/cambodia-steps-up-travel-curbs-on-foreigners">https://www.bangkokpost.com/world/1888300/cambodia-steps-up-travel-curbs-on-foreigners</a>
3	Bhutan	Full	<a href="https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Bhutan">https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Bhutan</a>
4	Luxembourg	Full	<a href="https://luxtimes.lu/luxembourg/40100-germany-border-checks-add-to-luxembourg-virus-lockdown">https://luxtimes.lu/luxembourg/40100-germany-border-checks-add-to-luxembourg-virus-lockdown</a>
5	Aruba	Full	<a href="https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Aruba">https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Aruba</a>

At the bottom, there are buttons for "PERSONAL HISTORY", "PROJECT HISTORY", and "SAVED QUERIES". The taskbar at the very bottom shows various application icons.

## CLUSTERING OF TABLES

This screenshot shows the Google Cloud Platform BigQuery interface with the same navigation bar and project context as the previous one. The main area displays a query editor window.

The "RUN" button is highlighted, and a note indicates that the query will process 15.5 KB when run. The query itself is a CREATE TABLE statement:

```
1 CREATE TABLE
2   tarkshyas-1636464439420.demo.lockdowndates_cleaned_partitioned_clustered
3   PARTITION BY
4     (news_date)
5   CLUSTER BY
6     Country_Region AS
7   SELECT
8     *
9   FROM
10  tarkshyas-1636464439420.demo.lockdowndates_cleaned
```

The "Processing location: US" is noted below the query. The "Query results" section is currently empty.

The bottom navigation bar and taskbar are identical to the first screenshot.

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with pinned projects and datasets. The main area shows a table named 'lockdowndate...' with the following schema:

Row	Country_Region	Type	Reference
1	Jordan	Full	<a href="https://www.nytimes.com/aponline/2020/03/21/world/middleeast/ap-ml-virus-outbreak-mideast.html">https://www.nytimes.com/aponline/2020/03/21/world/middleeast/ap-ml-virus-outbreak-mideast.html</a>
2	Burkina Faso	Full	<a href="https://en.wikipedia.org/wik/2020_coronavirus_pandemic_in_Burkina_Faso">https://en.wikipedia.org/wik/2020_coronavirus_pandemic_in_Burkina_Faso</a>
3	Sri Lanka	Full	<a href="https://www.deccanherald.com/internationl/china-offers-to-build-makehift-covid-19-hospitals-in-india-816718.html">https://www.deccanherald.com/internationl/china-offers-to-build-makehift-covid-19-hospitals-in-india-816718.html</a>
4	Taiwan	Full	<a href="https://en.wikipedia.org/wik/2020_coronavirus_pandemic_in_Taiwan">https://en.wikipedia.org/wik/2020_coronavirus_pandemic_in_Taiwan</a>
5	Albania	Full	<a href="https://en.wikipedia.org/wik/2020_coronavirus_pandemic_in_Albania">https://en.wikipedia.org/wik/2020_coronavirus_pandemic_in_Albania</a>
6	Estonia	Full	<a href="https://en.wikipedia.org/wik/2020_coronavirus_pandemic_in_Estonia">https://en.wikipedia.org/wik/2020_coronavirus_pandemic_in_Estonia</a>
7	Puerto Rico	Full	<a href="https://en.wikipedia.org/wik/2020_coronavirus_pandemic_in_Puerto_Rico">https://en.wikipedia.org/wik/2020_coronavirus_pandemic_in_Puerto_Rico</a>

Below the table, there are buttons for 'COMPOSE NEW QUERY', 'SHARE', 'COPY', 'SNAPSHOT', 'DELETE', and 'EXPORT'. The bottom status bar shows system information like battery level, signal strength, and date/time.

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with pinned projects and datasets. The main area shows a table named 'web\_analytics' with the following schema:

Field name	Type	Mode	Policy Tags	Description
visitorId	INTEGER	NULLABLE		
visitNumber	INTEGER	NULLABLE		
visitId	INTEGER	NULLABLE		
visitStartTime	INTEGER	NULLABLE		
date	STRING	NULLABLE		
totals	RECORD	NULLABLE		

Below the table, there are buttons for 'COMPOSE NEW QUERY', 'SHARE', 'COPY', 'SNAPSHOT', 'DELETE', and 'EXPORT'. A message at the bottom right says 'Created temp' and '10:45:03 AM GMT+5'. The bottom status bar shows system information like battery level, signal strength, and date/time.

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, the Explorer sidebar displays a project structure under 'tarkshyas-1636464439420'. A dataset named 'demo' is selected, containing tables like 'lockdowndates', 'lockdowndates\_cleaned', and 'lockdowndates\_cleaned\_partitioned'. The 'lockdowndates\_cleaned\_partitioned' table is currently previewed. The table has 5 rows and includes columns for 'Region' and 'Type'. The first row is a full dataset, while subsequent rows are partial datasets. The table is described as a partitioned table.

Row	Country_Region	Type	Reference
1	Barbados	Full	<a href="https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Barbados">https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Barbados</a>
2	Cambodia	Partial	<a href="https://www.bangkokpost.com/world/1888300/cambodia-steps-up-travel-curbs-on-foreigners">https://www.bangkokpost.com/world/1888300/cambodia-steps-up-travel-curbs-on-foreigners</a>
3	Bhutan	Full	<a href="https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Bhutan">https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Bhutan</a>
4	Luxembourg	Full	<a href="https://luxtimes.lu/luxembourg/40100-germany-border-checks-add-to-luxembourg-virus-lockdown">https://luxtimes.lu/luxembourg/40100-germany-border-checks-add-to-luxembourg-virus-lockdown</a>
5	Aruba	Full	<a href="https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Aruba">https://en.wikipedia.org/wiki/2020_coronavirus_pandemic_in_Aruba</a>

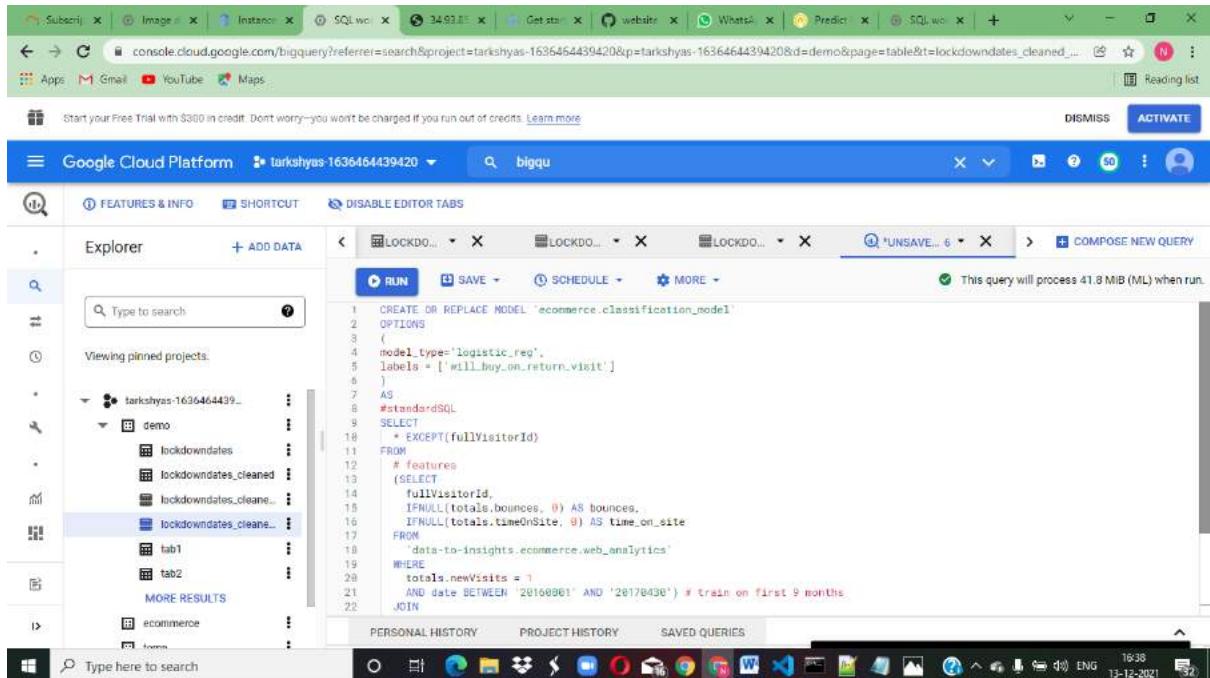
The screenshot shows the Google Cloud Platform BigQuery interface with the 'Create dataset' dialog box open. The dialog box allows setting up a new dataset. It includes fields for 'Project ID' (set to 'tarkshyas-1636464439420'), 'Dataset ID' (set to 'demo'), and 'Data location' (set to 'us (multiple regions in United States)'). There are also sections for 'Default table expiration' (with an unchecked checkbox for 'Enable table expiration') and 'Encryption' (with options for 'Google-managed encryption key' and 'Customer-managed encryption key (CMEK)'). At the bottom right of the dialog box are 'CREATE DATASET' and 'CANCEL' buttons.

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, the Explorer sidebar displays a project named 'tarkshyas-1636464439420' with a 'demo' dataset selected. Inside 'demo', there are tables like 'lockdowndates', 'lockdowndates\_cleaned', and 'tab1'. A search bar at the top right contains the query 'bigqu'. On the right, a 'Create dataset' dialog is open. The 'Project ID' is set to 'tarkshyas-1636464439420'. The 'Dataset ID' is set to 'temp'. The 'Data location' is set to 'us (multiple regions in United States)'. Under 'Default table expiration', there is a checkbox for 'Enable table expiration' which is unchecked. Below it is a 'Default maximum table age' field with a dropdown menu showing 'Days'. The 'Encryption' section has two options: 'Google-managed encryption key' (selected) and 'Customer-managed encryption key (CMK)' (unselected). A 'CREATE DATASET' button is at the bottom right of the dialog. The system status bar at the bottom right shows '16:35 13-12-2021'.

This screenshot is similar to the one above, showing the 'Create dataset' dialog for a new dataset named 'ecommerce'. The 'Project ID' is 'tarkshyas-1636464439420'. The 'Dataset ID' is 'ecommerce'. The 'Data location' is 'us (multiple regions in United States)'. The 'Default table expiration' checkbox is unchecked. The 'Encryption' section shows 'Google-managed encryption key' selected. In the background, a code editor window is visible with a complex SQL query. The query starts with a SELECT statement from a table named 'features' (which is an alias for 'SELECT fullvisitorid, IFNULL(totals.bounces, 0) AS bounces, IFNULL(totals.timeOnSite, 0) AS time\_on\_site FROM data-to-insights.ecommerce.web\_analytics'). It includes WHERE clauses for 'newVisits = 1' and 'date BETWEEN '20160801' AND '20170430''. It also includes a JOIN clause with another table 'totals' (SELECT fullvisitorid, IF(COUNTIF(transactions > 0) AND totals.transactions > 0, 1, 0) AS newVisits FROM data-to-insights.ecommerce.web\_analytics GROUP BY fullvisitorid) using the 'fullvisitorid' column. The system status bar at the bottom right shows '16:37 13-12-2021'.

## BIGQUERY ML MODELS

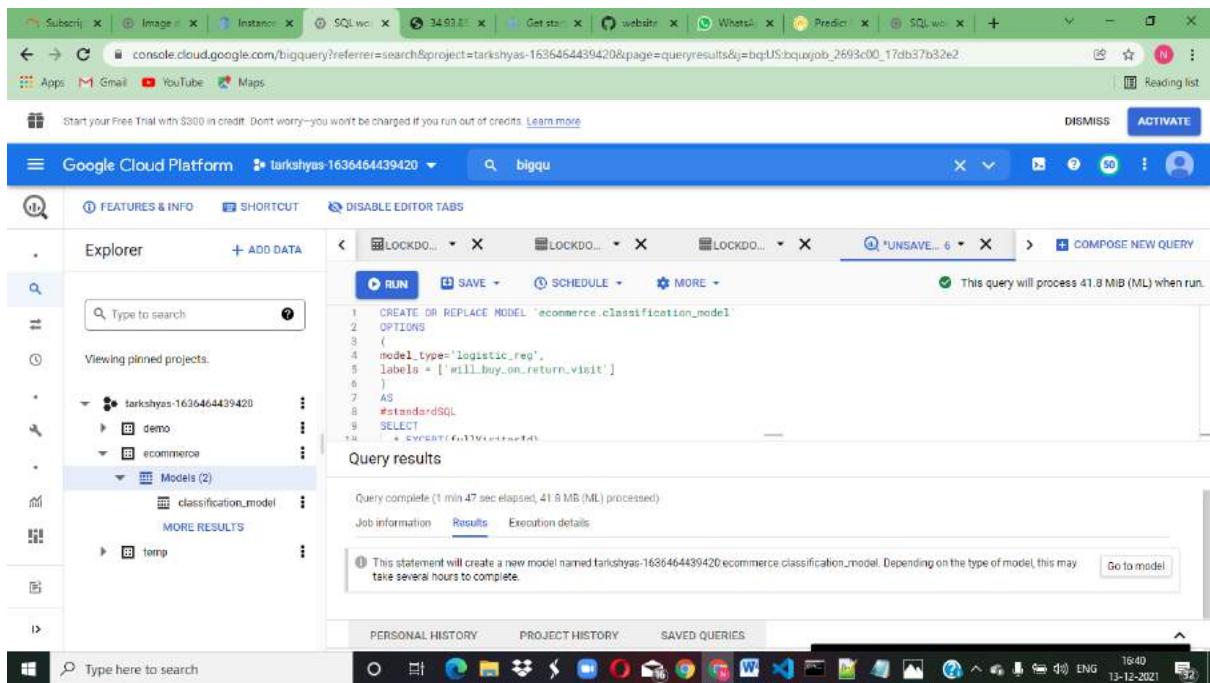
### Classification model 1



The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with a tree view of projects, datasets, and tables. In the main area, a query editor window is open with the following SQL code:

```
1 CREATE OR REPLACE MODEL `ecommerce.classification_model`
2 OPTIONS
3 (
4   model_type='logistic_reg',
5   labels = ['will_buy_on_return_visit']
6 )
7 AS
8 #standardSQL
9 SELECT
10   * EXCEPT(fullyVisitorId)
11 FROM
12   # features
13   (SELECT
14     fullVisitorId,
15     IFNULL(totals.bounces, 0) AS bounces,
16     IFNULL(totals.timeOnSite, 0) AS time_on_site
17   FROM
18     `data-to-insights.ecommerce.web_analytics`
19   WHERE
20     totals.newVisits = 1
21     AND date BETWEEN '20160801' AND '20170430') /* train on first 9 months
22 JOIN
```

The status bar at the bottom right indicates the query will process 41.8 MB (ML) when run.



The screenshot shows the Google Cloud Platform BigQuery interface after running the query. The main area displays the 'Query results' section with the following message:

Query complete (1 min 47 sec elapsed, 41.8 MB (ML) processed)

Job information Results Execution details

A note at the bottom states: "This statement will create a new model named tarkshyas-1636464439420.ecommerce.classification\_model. Depending on the type of model, this may take several hours to complete." A 'Go to model' button is also present.

The screenshot shows the Google Cloud Platform BigQuery interface. The main title bar says "Google Cloud Platform" and "tarkshyas-1636464439420". The search bar contains "bigqu". The left sidebar has an "Explorer" section with pinned projects: "tarkshyas-1636464439420" (demo, ecommerce, Models (2)), "temp", and "MORE RESULTS". The main content area shows a "classification\_model" card with tabs for "DETAILS", "TRAINING", "EVALUATION", and "SCHEMA". Under "DETAILS", it shows "Model type: LOGISTIC\_REGRESSION" and "Data location: US". Below this is a "Model Details" table with columns for Model ID, Description, Labels, Date created, Model expiration, Date modified, and Data location. The "Model ID" is "tarkshyas-1636464439420:ecommerce.classification\_model". The "Data location" is "US". At the bottom of the main area are tabs for "PERSONAL HISTORY", "PROJECT HISTORY", and "SAVED QUERIES". The status bar at the bottom right shows "16:41 13-12-2021".

The screenshot shows the Google Cloud Platform BigQuery interface. The main title bar says "Google Cloud Platform" and "tarkshyas-1636464439420". The search bar contains "bigqu". The left sidebar has an "Explorer" section with pinned projects: "tarkshyas-1636464439420" (demo, ecommerce, Models (2)), "temp", and "MORE RESULTS". The main content area shows a query editor with a code block containing SQL code for evaluating a model. The code includes selecting features from a table, using ML.EVALUATE to get model quality, and selecting features except for fullVisitorId. A note above the code says "This query will process 41.6 MB when run." The status bar at the bottom right shows "16:42 13-12-2021".

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with a pinned project 'tarkshyas-1636464439420'. Under 'Models (2)', the 'classification\_model' is selected. The main panel shows a query editor with the following SQL code:

```
#Evaluating model
SELECT
  roc_auc,
  CASE
    WHEN roc_auc > .9 THEN 'good'
  END AS model_quality

```

The status bar indicates 'Processing location: US'. Below the code, the 'Query results' section shows one row of data:

Row	roc_auc	model_quality
1	0.7238591408591408	decent

At the bottom, there are tabs for 'Job information', 'Results' (which is selected), 'JSON', and 'Execution details'.

## Classification model 2

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with a pinned project 'tarkshyas-1636464439420'. Under 'Models (2)', the 'classification\_model' is selected. The main panel shows a query editor with the following SQL code:

```
CREATE OR REPLACE MODEL `ecommerce.classification_model_2`
OPTIONS
  (nodeType='logistic_reg', labels = ['will_buy_on_return_visit'])
WITH all_visitor_stats AS (
  SELECT
    fullvisitorid,
    IF(COUNTIF(totals.transactions = 0 AND totals.newVisits IS NULL) > 0, 1, 0) AS will_buy_on_return_visit
  FROM `data-to-insights.ecommerce.web_analytics`
  GROUP BY fullvisitorid
)
# add in new features
SELECT * EXCEPT(unique_session_id) FROM (
  SELECT
    CONCAT(fullvisitorid, CAST(visitId AS STRING)) AS unique_session_id,
    # labels
    will_buy_on_return_visit,
    MAX(CAST(h.eCommerceAction.action_type AS INT64)) AS latest_ecommerce_progress,
    # behavior on the site
)
```

The status bar indicates 'Processing location: US'. Below the code, the 'Query results' section shows one row of data:

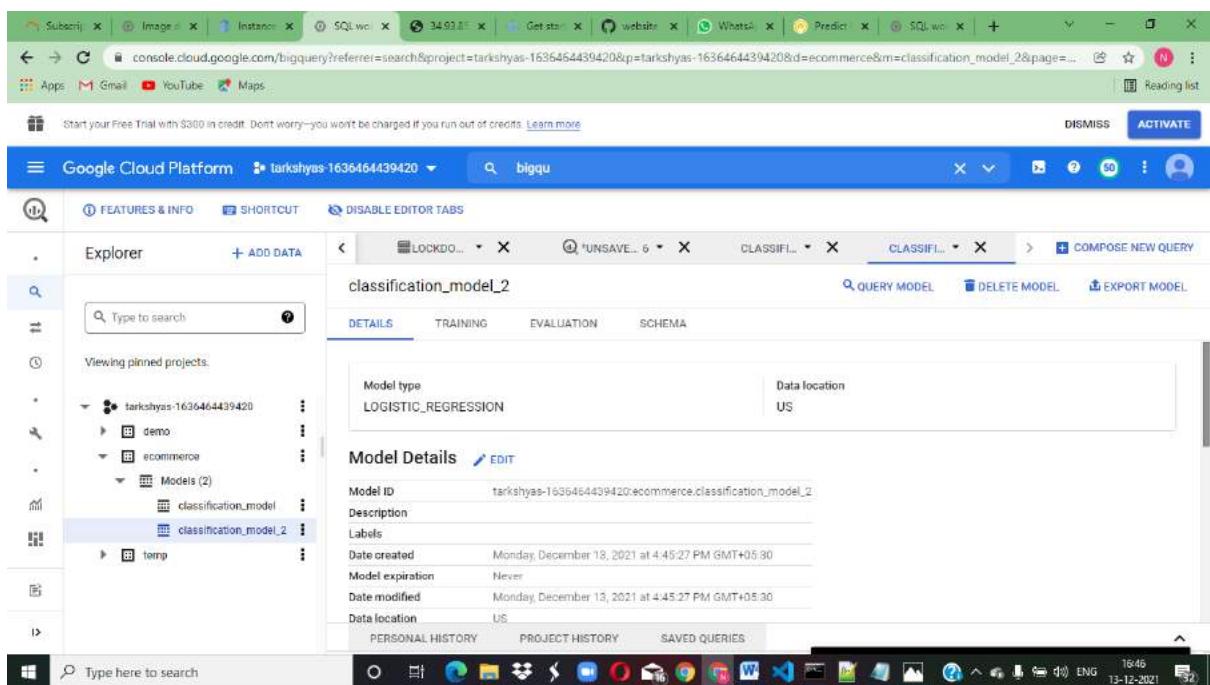
Row	unique_session_id	will_buy_on_return_visit	latest_ecommerce_progress
1	fullvisitorid.visitId	0	0

At the bottom, there are tabs for 'Job information', 'Results' (which is selected), 'JSON', and 'Execution details'.

The screenshot shows the Google Cloud Platform BigQuery interface. The top navigation bar includes tabs for Subscriptions, Images, Instances, SQL, 34.99, Get started, website, What's new, Predict, and SQL. A banner at the top encourages users to start a free trial with \$300 in credit. The main window has a blue header bar with the title "Google Cloud Platform" and the project ID "tarkshyas-1636464439420". Below the header is a toolbar with "FEATURES & INFO", "SHORTCUT", and "DISABLE EDITOR TABS". The left sidebar is titled "Explorer" and shows a tree view of projects, including "tarkshyas-1636464439420" which contains "demo", "ecommerce", and "Models (2)" which further contain "classification\_model" and "classification\_model\_2". The main content area displays a query editor with the following code:

```
31   UNNEST(hits) AS h
32   JOIN all_visitor_stats USING(fullvisitorid)
33   WHERE t=1
34   # only predict for new visits
35   AND totals.newVisits = 1
```

The status bar indicates "Processing location: US". Below the code editor is a section titled "Query results" with the message "Query complete (1 min 58 sec elapsed, 120 MB (ML) processed)". It includes tabs for "Job information", "Results" (which is selected), and "Execution details". A note states: "This statement will create a new model named tarkshyas-1636464439420 ecommerce classification\_model\_2. Depending on the type of model, this may take several hours to complete." At the bottom of the interface is a search bar and a taskbar with various icons.



The screenshot shows the Google Cloud Platform BigQuery interface. In the center, there is a code editor window titled 'RUN' containing a SQL script. The script uses ML.EVALUATE to create a model quality column based on ROC AUC values. Below the code editor is a 'Query results' section. The results table has two columns: 'Row' and 'model\_quality'. There is one row with the value 'good'. At the bottom of the interface, there is a navigation bar with tabs for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES'.

```
1 #standardSQL
2 SELECT
3     roc_auc,
4     CASE
5         WHEN roc_auc > .9 THEN 'good'
6         WHEN roc_auc > .8 THEN 'fair'
7         WHEN roc_auc > .7 THEN 'decent'
8         WHEN roc_auc > .6 THEN 'not great'
9         ELSE 'poor' END AS model_quality
10
11     FROM
12         ML.EVALUATE(MODEL ecommerce.classification_model_2,
13             WITH all_visitor_stats AS (
14                 SELECT
15                     fullvisitorid,
16                     IF(COUNTIF(totals.transactions > 0 AND totals.newVisits IS NULL) > 0, 1, 0) AS will_buy_on_return_visit
17                 FROM data-to-Insights.ecommerce.web_analytics
18             )
19             GROUP BY fullvisitorid
20         )
```

This screenshot shows the same Google Cloud Platform BigQuery interface as the previous one, but the results table now contains multiple rows. The 'model\_quality' column includes entries like 'good', 'fair', 'decent', and 'not great' corresponding to different ROC AUC values. The rest of the interface remains the same, with the code editor at the top and the results table at the bottom.

Row	model_quality
1	good
2	fair
3	decent
4	not great