

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-template>

STEP 1: Create a Docker Image

Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to **tarkshyas-1638768744958**.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
tsgcp user1@cloudshell: ~ (tarkshyas-1638768744958)\$ docker run -d -p 8080:80 nginx:latest
docker: Error response from daemon: driver failed programming external connectivity on endpoint vigorous_rhodes (1bab3e3db28c0579a92df134a62b3de97a8ala6821bf47fd26a3d084713ce2e): Bind for 0.0.0.0:8080 failed: port is already allocated.
tsgcp user1@cloudshell: ~ (tarkshyas-1638768744958)\$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
72f58ce9dc15 nginx:latest "docker-entrypoint..." 35 minutes ago Up 35 minutes 0.0.0.0:8080->80/tcp gracious_lumiere
tsgcp user1@cloudshell: ~ (tarkshyas-1638768744958)\$ git clone https://github.com/learning-zone/website-templates
fatal: destination path 'website-templates' already exists and is not an empty directory.
tsgcp user1@cloudshell: ~ (tarkshyas-1638768744958)\$ ls
creatable.yaml README-cloudshell.txt tarkshyas-1638768744958.sln
tsgcp user1@cloudshell: ~ (tarkshyas-1638768744958)\$ cd website-templates
tsgcp user1@cloudshell: ~/website-templates (tarkshyas-1638768744958)\$ cd above-educational-bootstrap-responsive-template
tsgcp user1@cloudshell: ~/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1638768744958)\$ ls
about.htm contact.htm courses.htm css fonts img index.htm js portfolio.htm pricing.htm readme.txt
tsgcp user1@cloudshell: ~/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1638768744958)\$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
72f58ce9dc15 nginx:latest "docker-entrypoint..." 37 minutes ago Up 37 minutes 0.0.0.0:8080->80/tcp gracious_lumiere
tsgcp user1@cloudshell: ~/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1638768744958)\$ docker cp about.htm 72f58ce9dc15:/usr/share/nginx/html/
tsgcp user1@cloudshell: ~/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1638768744958)\$ docker cp contact.htm 72f58ce9dc15:/usr/share/nginx/html/
tsgcp user1@cloudshell: ~/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1638768744958)\$ docker cp courses.htm 72f58ce9dc15:/usr/share/nginx/html/
tsgcp user1@cloudshell: ~/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1638768744958)\$ docker cp fonts 72f58ce9dc15:/usr/share/nginx/html/
tsgcp user1@cloudshell: ~/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1638768744958)\$ docker cp js 72f58ce9dc15:/usr/share/nginx/html/
tsgcp user1@cloudshell: ~/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1638768744958)\$ docker cp portfolio.htm 72f58ce9dc15:/usr/share/nginx/html/
tsgcp user1@cloudshell: ~/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1638768744958)\$ docker cp pricing.htm 72f58ce9dc15:/usr/share/nginx/html/
tsgcp user1@cloudshell: ~/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1638768744958)\$ docker cp readme.txt 72f58ce9dc15:/usr/share/nginx/html/
tsgcp user1@cloudshell: ~/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1638768744958)\$ docker cp testweb3:latest 72f58ce9dc15:
sha256:7e4c103bdfe04b7427bd7b5c29e249f23f3feeb455c62616ed639a6b6be
tsgcp user1@cloudshell: ~/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1638768744958)\$ docker images
tsgcp user1@cloudshell: ~

Subscription Details | Nuverro | Revisions – testweb3 – Cloud Run | Images – Container Registry – tarkshyas-1638768744958 | VM Instances – Compute Engine | +

Google Cloud Platform | tarkshyas-1638768744958 | conta

Container Registry | Images | DELETE | LEARN

Images

testweb3

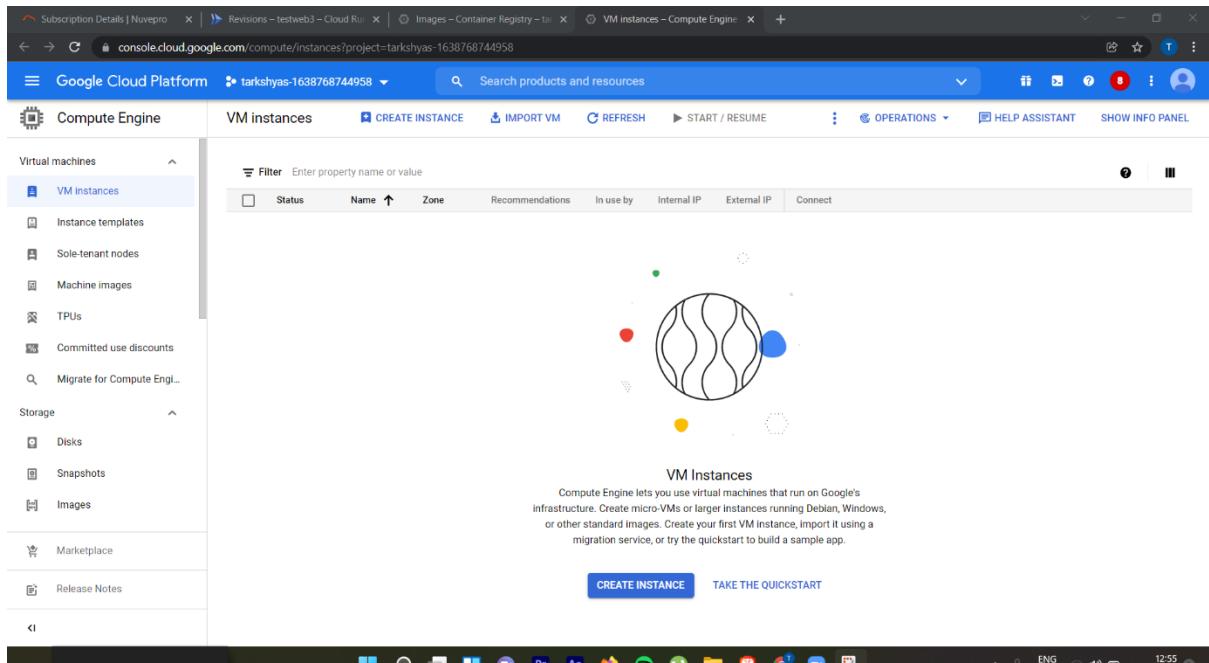
Name Tags Virtual Size Created Uploaded

2ac8d214873 latest 55.5 MB 22 minutes ago 16 minutes ago

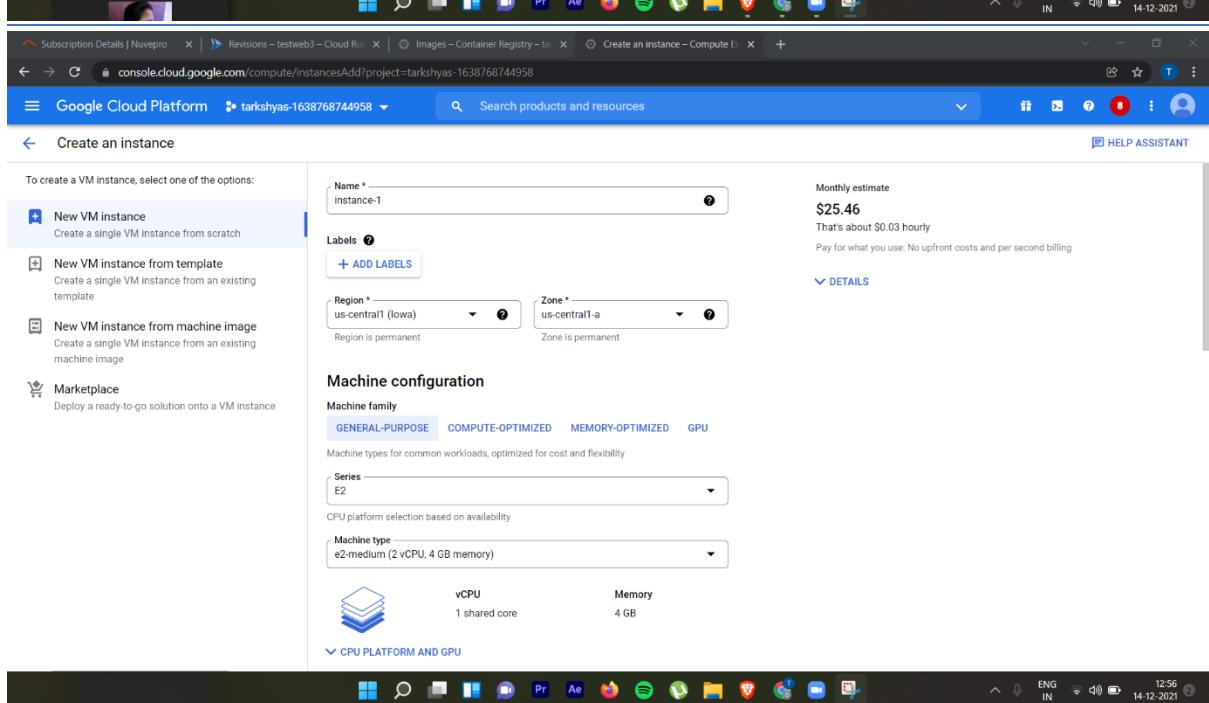
Marketplace

Release Notes

STEP 2: Create a VM instance



The screenshot shows the Google Cloud Platform Compute Engine VM Instances page. On the left, there's a sidebar with options like VM instances, Instance templates, Sole-tenant nodes, Machine images, TPUs, Committed use discounts, and Migrate for Compute Eng... Under Storage, it lists Disks, Snapshots, and Images. There are also Marketplace and Release Notes sections. The main area displays a globe icon with colored dots (green, red, blue, yellow) and a "VM Instances" section with a brief description of what Compute Engine lets you do. At the bottom are "CREATE INSTANCE" and "TAKE THE QUICKSTART" buttons.



The screenshot shows the "Create an instance" wizard, step 1: Machine configuration. It starts with a sidebar with options: New VM instance (selected), New VM instance from template, New VM instance from machine image, and Marketplace. The main form has fields for Name (instance-1), Labels (+ ADD LABELS), Region (us-central1 (Iowa)), Zone (us-central1-a), and Monthly estimate (\$25.46). It also includes a "DETAILS" section with a note about pay-as-you-go billing. Below this is the "Machine configuration" section with tabs for GENERAL-PURPOSE, COMPUTE-OPTIMIZED, MEMORY-OPTIMIZED, and GPU. Under GENERAL-PURPOSE, it shows Series (E2), vCPU (1 shared core), and Memory (4 GB). A "CPU PLATFORM AND GPU" section is partially visible at the bottom.

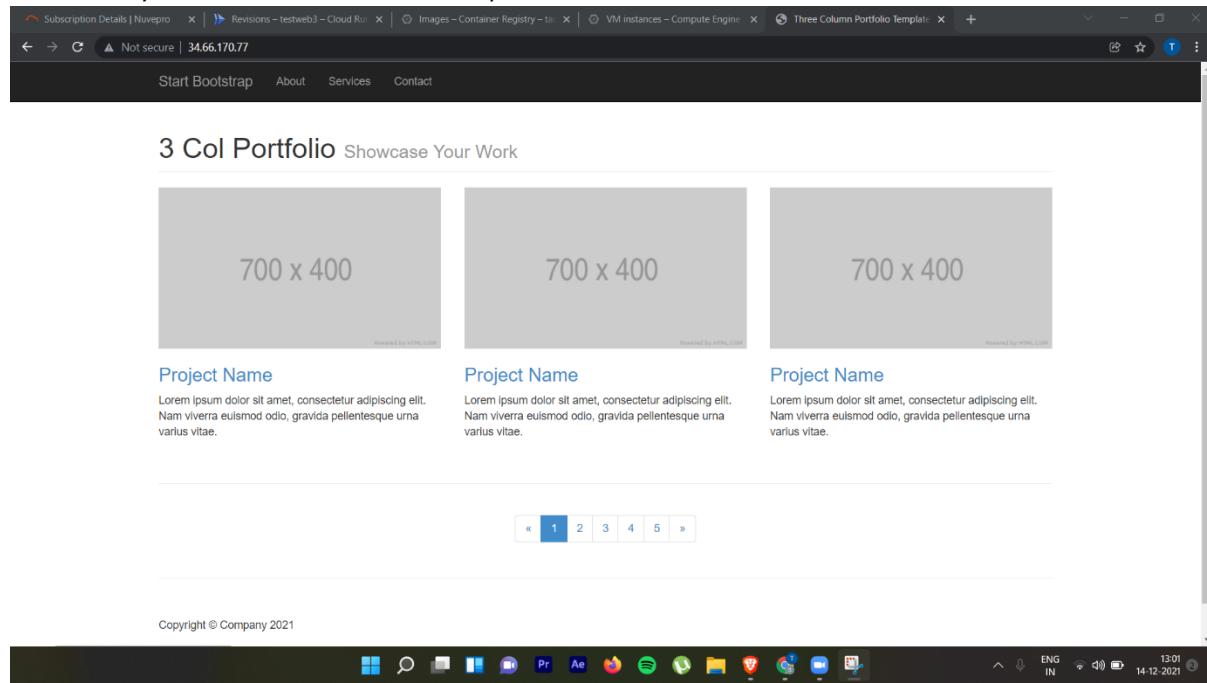
STEP 3: Deploy the Container

The screenshot shows two consecutive screenshots of the Google Cloud Platform interface.

Top Screenshot: The "Create an instance" dialog for deploying a container. The "Container Image" field is set to "us.gcr.io/tarkshyas-1638768744958/testweb3". The "Restart policy" is set to "Always". Under "Arguments", there is a "+ ADD ARGUMENT" button. Under "Volume mounts", there is a "+ ADD VOLUME" button. At the bottom are "SELECT" and "CANCEL" buttons.

Bottom Screenshot: The "VM instances" page under the Compute Engine section. The left sidebar shows "Virtual machines" with "VM Instances" selected. The main table lists one instance named "Instance-1" in the "us-central1-a" zone. The table columns include Status, Name, Zone, Recommendations, In use by, Internal IP, External IP, and Connect. The "Status" column for "Instance-1" shows a green checkmark. The "Connect" column for "Instance-1" has an "SSH" dropdown menu.

Successfully Created VM and host a sample website



2.Create a VPC network to communicate between 2 VM instances present in different networks.

STEP 1: Create a 2 VPC Networks

The screenshot shows two separate sessions of the Google Cloud Platform VPC network creation interface.

Session 1 (Top): Creating VPC network 'vpc1'

- Subnet creation mode:** Custom (selected)
- New subnet:**
 - Name:** us-central1
 - Description:** (empty)
 - Region:** us-central1
 - IP address range:** 10.0.0.0/9
- Private Google Access:** Off (selected)
- Flow logs:** A progress dialog box is visible: "Creating network 'vpc1'..."

Session 2 (Bottom): Creating VPC network 'vpc2'

- Subnet creation mode:** Custom (selected)
- New subnet:**
 - Name:** asia-south-1
 - Description:** (empty)
 - Region:** asia-south1
- Success message:** "Successfully created network 'vpc1'."

2 VPCs Created (vpc1 in us-central-1 and vpc2 in asia-south-1)

The screenshot shows the Google Cloud Platform VPC networks page. On the left, there's a sidebar with 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 area displays a table of VPC networks. The table has columns for Name, Region, Subnets, MTU, Mode, IP address ranges, Gateways, Firewall Rules, Global dynamic routing, and Flow logs. It lists 'default' (Region: us-central1, Subnets: 29, MTU: 1460, Mode: Auto) and 'vpc1' (Region: us-central1, Subnets: 1, MTU: 1460, Mode: Custom). Below 'vpc1' is another row for 'vpc2' (Region: asia-south1, Subnets: 1, MTU: 1460, Mode: Custom). A success message 'Successfully created network "vpc2"' is shown in a toast at the bottom. The status bar at the bottom right shows ENG IN, 09:54, and 14-12-2021.

STEP 2: Enable Firewall rule for both vpc1 and vpc2

The screenshot shows the Google Cloud Platform VPC network details page for 'vpc1'. The sidebar includes 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 area shows 'vpc1' with settings for Subnet creation mode (Custom subnets), Dynamic routing mode (Regional), DNS server policy (None), and Maximum transmission unit (1460). Below these are tabs for SUBNETS, STATIC INTERNAL IP ADDRESSES, FIREWALL POLICIES, FIREWALL RULES, ROUTES, VPC NETWORK PEERING, and PRIVATE SERVICE CONNECTION. The FIREWALL RULES tab is selected. A table header for 'ADD FIREWALL RULE' is visible. The status bar at the bottom right shows ENG IN, 09:54, and 14-12-2021.

Sreekanth Santhosh-198642

The screenshot shows two side-by-side instances of the Google Cloud Platform Firewall rule creation interface. Both instances are for a VPC network named 'vpc1'.

Left Instance (Top):

- Name:** firewall-rule1
- Description:** (Empty)
- Logs:** Off
- Network:** vpc1
- Priority:** 1000
- Direction of traffic:** Ingress
- Action on match:** Allow

Right Instance (Bottom):

- Action on match:** Allow
- Targets:** All instances in the network
- Source filter:** IPv4 ranges
- Source IPv4 ranges:** 0.0.0.0/0
- Second source filter:** None
- Protocols and ports:** Allow all

Both instances include a 'CREATE' button at the bottom right and an 'EQUIVALENT COMMAND LINE' dropdown menu.

STEP 3: Create 2 VMs

The screenshot shows the Google Cloud Platform 'Create an instance' interface. On the left, a sidebar lists options: 'New VM instance' (selected), 'New VM instance from template', 'New VM instance from machine image', and 'Marketplace'. The main form is titled 'Create an instance' and includes fields for 'Name' (set to 'us-central-vm'), 'Labels' (empty), 'Region' (set to 'us-central1 (Iowa)'), 'Zone' (set to 'us-central1-c'), and 'Machine configuration' (set to 'GENERAL-PURPOSE' with 'Series' as 'E2' and 'Machine type' as 'e2-medium (2 vCPU, 4 GB memory)'). A monthly estimate of '\$25.46' is shown, along with a note that it's about \$0.03 hourly. The bottom of the screen shows a taskbar with various application icons.

Sreekanth Santhosh-198642

The screenshot shows two windows of the Google Cloud Platform interface.

Top Window: Create an instance

To create a VM instance, select one of the options:

- New VM instance**: Create a single VM instance from scratch.
- New VM instance from template**: Create a single VM instance from an existing template.
- New VM instance from machine image**: Create a single VM instance from an existing machine image.
- Marketplace**: Deploy a ready-to-go solution onto a VM instance.

Network interfaces

Network interface is permanent.

Edit network interface

Network: vpc1
Subnetwork: us-central1 (10.0.0.0/9)
Primary internal IP: Ephemeral (Automatic)

Alias IP ranges

External IP: Ephemeral

Network Service Tier

Premium (selected) or Standard (us-central1)

Public DNS PTR Record

Monthly estimate: \$25.46 (That's about \$0.03 hourly). Pay for what you use: No upfront costs and per second billing.

DETAILS

Bottom Window: VM instances

Compute Engine

VM instances

Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
Green	asia-south-vm	asia-south1-c			10.160.0.2 (nic0)	34.93.125.223	SSH
Green	us-central-vm	us-central1-c			10.0.0.2 (nic0)	34.72.145.125	SSH

STEP 4: Go to External IP addresses in VPC Network

The screenshot shows the Google Cloud Platform interface for creating a static external IP address. The left sidebar is titled 'VPC network' and lists options like 'VPC networks', 'External IP addresses' (which is selected), 'Bring your own IP', 'Firewall', 'Routes', 'VPC network peering', 'Shared VPC', 'Serverless VPC access', and 'Packet mirroring'. The main panel is titled 'Reserve a static address' and contains fields for 'Name' (set to 'us-central1'), 'Description', 'Network Service Tier' (set to 'Premium'), 'IP version' (set to 'IPv4'), 'Type' (set to 'Regional'), 'Region' (set to 'us-central1 (Iowa)'), and 'Attached to' (set to 'None'). A note at the bottom states: 'Some of the instances may be disabled due to the "External IPs for VM instances" organization policy. [Learn more](#)'. Below this, a warning message says: 'Static IP addresses not attached to an instance or load balancer are billed at a higher hourly rate. [Pricing details](#)'.

The screenshot shows the Google Cloud Platform interface for managing external IP addresses. The left sidebar is the same as the previous screen. The main panel is titled 'External IP addresses' and displays a table of existing addresses. The table columns are: Name, External Address, Region, Type, Version, In use by, Network Tier, and Labels. The table shows two rows: one for 'asia-south-1' (IP 34.93.239.85) which is 'None' and 'Premium' tier, and one for 'us-central1' (IP 34.68.136.149) which is also 'None' and 'Premium' tier. A new row is added to the table for the recently created address: 'us-central1' (IP 34.72.145.125), which is 'Ephemeral' and 'IPv4', and is currently 'In use by' VM instance 'us-central-vm' (Zone us-central1-c). A success message at the top right says 'Successfully created address "us-central1".'

STEP 5: Create a Classic VPN

The screenshot shows the Google Cloud Platform Hybrid Connectivity interface. On the left sidebar, there are options for VPN, Interconnect, Cloud Routers, and Network Connectivity Center. The main content area is titled "Create a VPN" and displays two network connection options:

- High-availability (HA) VPN**: Supports dynamic routing (BGP) only. It shows two routers connected via a BGP session, with each router having two tunnels (Tunnel 1 and Tunnel 2) connecting to a VPC network.
- Classic VPN**: Supports dynamic routing and static routing. It shows a single tunnel (Tunnel 1) connecting an On-premise network to a VPC network.

Below the diagrams are "CONTINUE" and "CANCEL" buttons. The status bar at the bottom indicates "10:07 14-12-2021".

The screenshot shows the "Create a VPN connection" page. The "Google Compute Engine VPN gateway" section includes fields for:

- Name ***: vpn-us-1
- Description**: (empty)
- Network ***: vpc1
- Region ***: us-central1 (Iowa)
- IP address ***: us-central1

The "Tunnels" section indicates you can have multiple tunnels to a single Peer VPN gateway. A "New item" button is visible. The status bar at the bottom indicates "10:08 14-12-2021".

The screenshot shows two stacked windows from the Google Cloud Platform Hybrid Connectivity interface.

Top Window: Create a router

- Left Sidebar:** Hybrid Connectivity, VPN, Interconnect, Cloud Routers, Network Connectivity Center.
- Form Fields:**
 - Name: us-router
 - Description: (empty)
 - Network: VPC Network: vpc1
 - Region: us-central1 (Iowa)
 - Google ASN: 64513
 - BGP peer keepalive interval: seconds (set to 30)
- Buttons:** CREATE, CANCEL, EQUIVALENT COMMAND LINE

Bottom Window: Create BGP session

- Left Sidebar:** Hybrid Connectivity, VPN, Interconnect, Cloud Routers, Network Connectivity Center.
- Form Fields:**
 - IKE version: IKEV2
 - IKE pre-shared key: fEzxwQAPSSzlaTMmpm2g8nYxg/2
 - Name: us-bgp
 - Peer ASN: 64514
 - Advertised route priority (MED): (empty)
 - Cloud Router BGP IP: 169.254.0.1
 - BGP peer IP: 169.254.0.2
 - BGP peer: Enabled (radio button selected)
- Buttons:** CREATE, CANCEL, ADD TUNNEL

Sreekanth Santhosh-198642

The screenshot displays two side-by-side configurations for creating a VPN connection in the Google Cloud Platform Hybrid Connectivity section.

Top Window (Left): Create a VPN connection

- IKE version:** IKEv2
- IKE pre-shared key:** fEzxwQAPSszlaTMmpm2g8nYXg/2d4Iug
- Cloud Router:** us-router
- BGP session:** us-bgp (IP: 169.254.0.1 Peer IP: 169.254.0.2)

Bottom Window (Right): Create a VPN connection

- New item:** vpn-2-tunnel-2
- Description:** (empty)
- Remote peer IP address:** 34.68.136.149
- IKE version:** IKEv2
- IKE pre-shared key:** fEzxwQAPSszlaTMmpm2g8nYXg/2d4Iug
- Cloud Router:** (dropdown menu)

Sreekanth Santhosh-198642

The screenshot shows two windows from the Google Cloud Platform Hybrid Connectivity interface.

Top Window: Edit BGP session

This window is titled "Edit BGP session". It contains the following fields:

- Name: asia-bgp
- Peer ASN: 64513
- Advertised route priority (MED): (disabled)
- Cloud Router BGP IP: 169.254.0.2
- BGP peer IP: 169.254.0.1
- BGP peer status: Enabled (radio button selected)

Bottom Window: VPN - Hybrid Connectivity

This window lists "CLOUD VPN TUNNELS", "CLOUD VPN GATEWAYS", and "PEER VPN GATEWAYS".

CLOUD VPN GATEWAYS

Gateway name	IP address	VPC network	Region	VPN tunnels	Description	Labels
vpn-asia-1	34.93.239.85	vpc2	asia-south1	vpn-2-tunnel-2		ADD VPN TUNNEL
vpn-us-1	34.68.136.149	vpc1	us-central1	vpn-1-tunnel-1		ADD VPN TUNNEL

The VPN tunnels are successfully connected and BGP sessions are also established

Tunnel name	Cloud VPN gateway (IP)	Peer VPN gateway (IP)	Cloud Router BGP IP	BGP Peer IP	Routing type	VPN tunnel status	BGP session st
vpn-1-tunnel-1 (Classic)	34.68.136.149	34.93.239.85	169.254.0.1	169.254.0.2	Dynamic (BGP)	Established	BGP established
vpn-2-tunnel-2 (Classic)	34.68.136.149	34.93.239.85	169.254.0.2	169.254.0.1	Dynamic (BGP)	Established	BGP established


```

ssh.cloud.google.com/projects/tarkshyas-1638768737406/zones/asia-south1-c/instances/asia-south-vm=authuser...
Connected, host fingerprint: ssh-rsa 0 AE:64:2E:C0:7B:A:FB:5C:61:5E:8F:40:0:CE
Linux asia-south-vm 4.19.0-18-cloud-amd64 #1 SMP Debian 4.19.208-1 (2021-09-29) 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.

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

root@tsqcp:~# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=62 time=251 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=62 time=251 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=62 time=251 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=62 time=251 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=62 time=251 ms
root@tsqcp:~# ping 10.160.0.2
PING 10.160.0.2 (10.160.0.2) 56(84) bytes of data.
64 bytes from 10.160.0.2: icmp_seq=1 ttl=62 time=259 ms
64 bytes from 10.160.0.2: icmp_seq=2 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=3 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=4 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=5 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=6 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=7 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=8 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=9 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=10 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=11 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=12 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=13 ttl=62 time=252 ms
64 bytes from 10.160.0.2: icmp_seq=14 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=15 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=16 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=17 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=18 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=19 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=20 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=21 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=22 ttl=62 time=251 ms
64 bytes from 10.160.0.2: icmp_seq=23 ttl=62 time=251 ms

```

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

<https://github.com/amunategui/rapid-prototyping-app-engine-yelp>

STEP 1: Create App Engine

Sreekanth Santhosh-198642

The image consists of two vertically stacked screenshots of the Google Cloud Platform (GCP) App Engine interface.

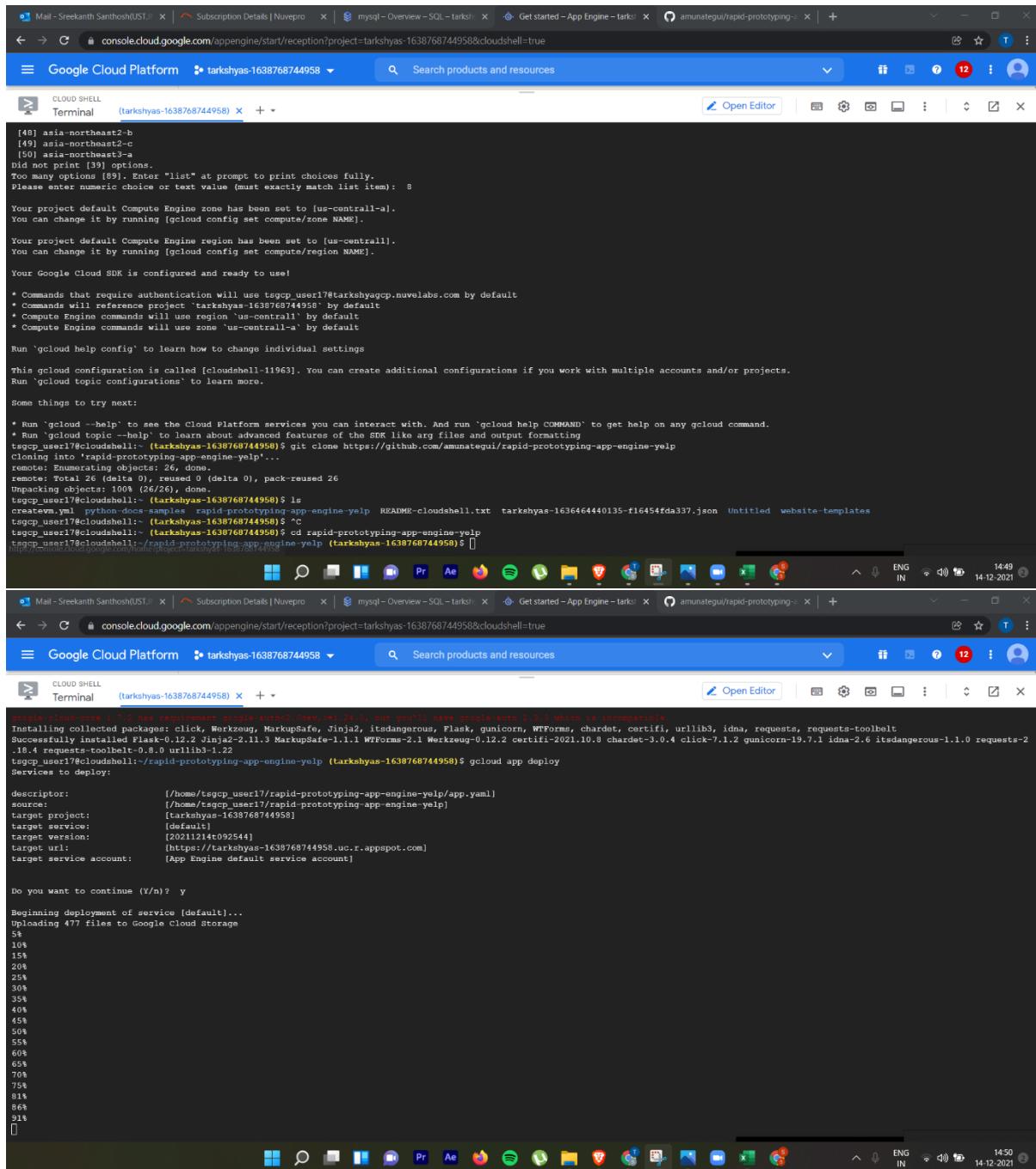
Top Screenshot: This shows the main 'Welcome to App Engine' page. The left sidebar is titled 'App Engine' and includes links for Dashboard, Services, Versions, Instances, Task queues, Cron jobs, Security scans, Firewall rules, Quotas, Memcache, Search, and Settings. Below the sidebar is a 'Release Notes' section. The main content area features a large 'Welcome to App Engine' heading, a sub-headline 'Build scalable apps in any language on Google's infrastructure', and a prominent blue 'CREATE APPLICATION' button.

Bottom Screenshot: This shows the 'Create app' wizard at the 'Select a location' step. The left sidebar is identical to the top one. The main content area has a title '1 Select a location — 2 Get started' and a 'Region' section with the sub-instruction 'Region is permanent.' It features a world map with red location markers and labels for regions like North America, South America, Europe, Africa, Asia, and Oceania. Below the map is a dropdown menu labeled 'Select a region *' with 'us-central' selected. A blue 'NEXT' button is at the bottom of this section. To the right of the main content is a sidebar with 'Recommended for you' sections, including 'App Engine Quickstart' and 'App Engine overview'.

Sreekanth Santhosh-198642

The screenshot shows the Google Cloud Platform App Engine dashboard. On the left, there's a sidebar with options like Dashboard, Services, Versions, Instances, Task queues, and Release Notes. The main area is titled 'Get started' under 'Resources'. It shows 'Language: Python' and 'Environment: Standard'. Below this, there are links to 'Read App Engine Python Standard Environment Documentation' and 'Visit Github' for code samples. To the right, there's a section titled 'Deploy with Google Cloud SDK' with a 'DOWNLOAD THE CLOUD SDK' button. It includes commands for 'Initialize your SDK' (\$ gcloud init) and 'Deploy to App Engine' (\$ gcloud app deploy). At the bottom, a terminal window is open with the command: \$ sudo apt-get install google-cloud-sdk-app-engine-python. The terminal also displays standard apt-get output about package lists and dependency trees. The status bar at the bottom right shows the date and time as 14-12-2021 13:20.

Sreekanth Santhosh-198642



```
[48] asia-northeast2-a
[49] asia-northeast2-b
[50] asia-northeast3-a
Did not print [39] options.
Too many options [89]. Enter "list" at prompt to print choices fully.
Please enter numeric choice or text value (must exactly match list item): 8
Your project default Compute Engine zone has been set to [us-central1-a].
You can change it by running (gcloud config set compute/zone NAME).

Your project default Compute Engine region has been set to [us-central1].
You can change it by running (gcloud config set compute/region NAME).

Your Google Cloud SDK is configured and ready to use!

* Commands that require authentication will use tsgcp_user17@tarkshyas-1638768744958 by default
* Commands will reference project 'tarkshyas-1638768744958' by default
* Compute Engine commands will use region 'us-central1' by default
* Compute Engine commands will use zone 'us-central1-a' by default

Run 'gcloud help config' to learn how to change individual settings

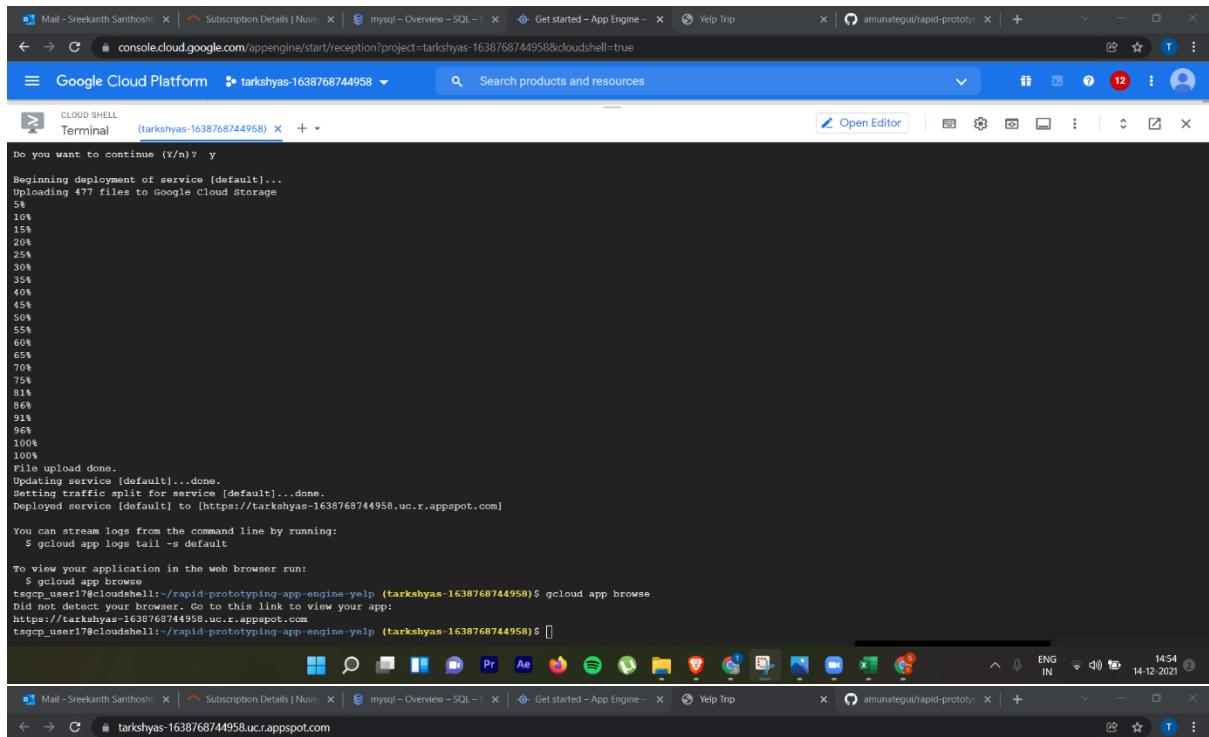
This gcloud configuration is called [cloudshell-11963]. You can create additional configurations if you work with multiple accounts and/or projects.
Run 'gcloud topic configurations' to learn more.

Some things to try next:

* Run 'gcloud --help' to see the Cloud Platform services you can interact with. And run 'gcloud help COMMAND' to get help on any gcloud command.
* Run 'gcloud topic --help' to learn about advanced features of the SDK like arg files and output formatting
tsgcp_user17@cloudshell:~$ git clone https://github.com/amonategu/rapid-prototyping-app-engine-yelp
Cloning into 'rapid-prototyping-app-engine-yelp'...
remote: Enumerating objects: 26, done.
remote: Total 26 (delta 0), reused 0 (delta 0), pack-reused 26
Unpacking objects: 100% (26/26), done.
tsgcp_user17@cloudshell:~$ ls
createvms.yaml python-docs-samples rapid-prototyping-app-engine-yelp README-cloudshell.txt tarkshyas-1638768744958/r16454fda337.json Untitled website-templates
tsgcp_user17@cloudshell:~$ ^C
tsgcp_user17@cloudshell:~$ cd rapid-prototyping-app-engine-yelp
tsgcp_user17@cloudshell:~/rapid-prototyping-app_engine_yelp (tarkshyas-1638768744958)$

google-cloud-core 1.7.0 has requirement google-auth==0.14.0, but you'll have google-auth 2.3.0 which is incompatible.
Installing collected packages: click, Werkzeug, MarkupSafe, Jinja2, itsdangerous, Flask, gunicorn, WTForms, chardet, certifi, urllib3, idna, requests, requests-toolbelt
Successfully installed Flask-0.12.2 Jinja2-2.11.3 MarkupSafe-1.1.1 WTForms-2.1 Werkzeug-0.12.2 certifi-2021.10.8 chardet-3.0.4 click-7.1.2 gunicorn-19.7.1 idna-2.6 itsdangerous-1.1.0 requests-2
.18.4 requests-toolbelt-0.8.0 urllib3-1.22
tsgcp_user17@cloudshell:~/rapid-prototyping-app-engine-yelp (tarkshyas-1638768744958)$ gcloud app deploy
Services to deploy:
descriptor:      [/home/tsgcp_user17/rapid-prototyping-app-engine-yelp/app.yaml]
source:          [/home/tsgcp_user17/rapid-prototyping-app-engine-yelp]
target project:  [tarkshyas-1638768744958]
target service:  [default]
target version:  [20211214t092544]
target url:      [https://tarkshyas-1638768744958.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 477 files to Google Cloud Storage
5%
10%
15%
20%
25%
30%
35%
40%
45%
50%
55%
60%
65%
70%
75%
80%
85%
90%
95%
```



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

Beginning deployment of service [default]...
Uploading 477 files to Google cloud Storage
5%
10%
15%
20%
25%
30%
35%
40%
45%
50%
55%
60%
65%
70%
75%
81%
86%
91%
96%
100%
100%
File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [<https://tarkshyas-1638768744958.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
tagcp_user17@cloudshell:/~/rapid-prototyping-app-engine-yelp (tarkshyas-1638768744958)\$ gcloud app browse
Did not detect your browser. Go to this link to view your app:
<https://tarkshyas-1638768744958.uc.r.appspot.com>
tagcp_user17@cloudshell:/~/rapid-prototyping-app-engine-yelp (tarkshyas-1638768744958)\$

Plan Your Trip

Departure: Destination: Business of Interest:

Enter start location Enter ending location Looking for what?



4. Host a website given below using App engine flexible environment

https://github.com/GoogleCloudPlatform/python-docs-samples/tree/main/appengine/flexible/django_cloudsql

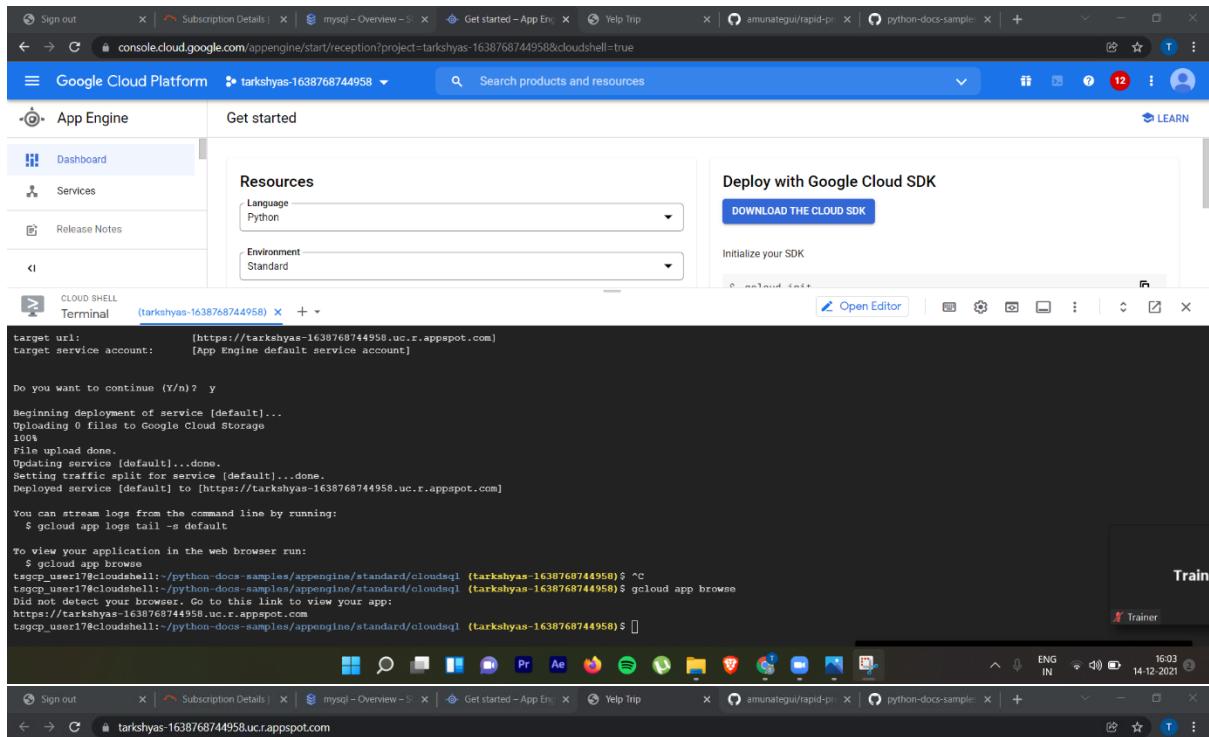
The screenshot shows the Google Cloud Platform App Engine dashboard for the project 'tarkshyas-1638768744958'. The left sidebar has 'App Engine' selected. The main area has 'Get started' selected. A 'Resources' section on the left shows 'Language: Python' and 'Environment: Standard'. On the right, there's a 'Deploy with Google Cloud SDK' section with a 'DOWNLOAD THE CLOUD SDK' button and an 'Initialize your SDK' section containing the command '\$ gcloud init'. Below these are tabs for 'CLOUD SHELL' and 'Terminal'. The 'Terminal' tab is active, showing deployment logs:

```
(hello_world) tsgcp user17@cloudshell: [tarkshyas-1638768744958]$ ls
createvms envs python-docs-samples rapid-prototyping-app-engine-yelp README-cloudshell.txt tarkshyas-163646440135-f1645fda337.json Untitled website-templates
(hello_world) tsgcp user17@cloudshell: [tarkshyas-1638768744958]$ cd python-docs-samples/appengine/flexible/django_cloudsql/
(hello_world) tsgcp user17@cloudshell: /python-docs-samples/appengine/flexible/django_cloudsql [tarkshyas-1638768744958]$ ls
app.yaml manage.py mysite nofile_config.py polls README.md requirements-test.txt requirements.txt
(hello_world) tsgcp user17@cloudshell: /python-docs-samples/appengine/flexible/django_cloudsql [tarkshyas-1638768744958]$ pip install -t lib -r requirements.txt
Collecting Django==3.2.9
  Downloading Django-3.2.9-py3-none-any.whl (7.9 MB)
    7.9 MB 13.4 MB/s
Collecting unicorn==20.1.0
  Downloading unicorn-20.1.0-py3-none-any.whl (79 kB)
    79 kB 7.3 MB/s
Collecting psycopg2-binary==2.9.2
  Downloading psycopg2_binary-2.9.2-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.0 MB)
    3.0 MB 73.8 MB/s
Collecting django-environ==0.8.1
```

The status bar at the bottom indicates the terminal was active at 15:50 on 14-12-2021.

This screenshot is identical to the one above, showing the same deployment logs in the terminal. A 'Trainer' label is visible in the bottom right corner of the interface.

Sreekanth Santhosh-198642



The screenshot shows the Google Cloud Platform App Engine dashboard. On the left, there's a sidebar with 'App Engine' selected, showing 'Dashboard', 'Services', and 'Release Notes'. The main area is titled 'Get started' with a 'Resources' section for Python and Standard environments. To the right, there's a 'Deploy with Google Cloud SDK' section with a 'DOWNLOAD THE CLOUD SDK' button and an 'Initialize your SDK' link. Below these is a terminal window titled 'CLOUD SHELL Terminal (tarkshyas-1638768744958)'. The terminal output shows the deployment process:

```
target url: [https://tarkshyas-1638768744958.uc.r.appspot.com]
target service account: [App Engine default service account]

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

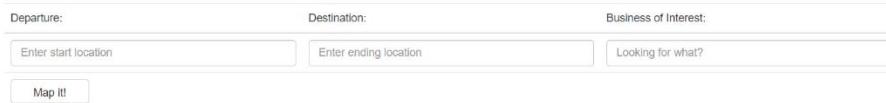
Beginning deployment of service [default]...
Uploading 0 files to Google Cloud Storage
100%
File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://tarkshyas-1638768744958.uc.r.appspot.com]

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
tagcp_user17@cloudshell:~/python-docs-samples/appengine/standard/cloudsql (tarkshyas-1638768744958)$ ^C
tagcp_user17@cloudshell:~/python-docs-samples/appengine/standard/cloudsql (tarkshyas-1638768744958)$ gcloud app browse
Did not detect your browser. Go to this link to view your app:
https://tarkshyas-1638768744958.uc.r.appspot.com
tagcp_user17@cloudshell:~/python-docs-samples/appengine/standard/cloudsql (tarkshyas-1638768744958)$
```

The taskbar at the bottom shows various open tabs and the current URL: tarkshyas-1638768744958.uc.r.appspot.com.

Plan Your Trip



The screenshot shows the 'Plan Your Trip' application interface. It has three input fields: 'Departure:' (with placeholder 'Enter start location'), 'Destination:' (with placeholder 'Enter ending location'), and 'Business of Interest:' (with placeholder 'Looking for what?'). Below these fields is a 'Map it!' button.



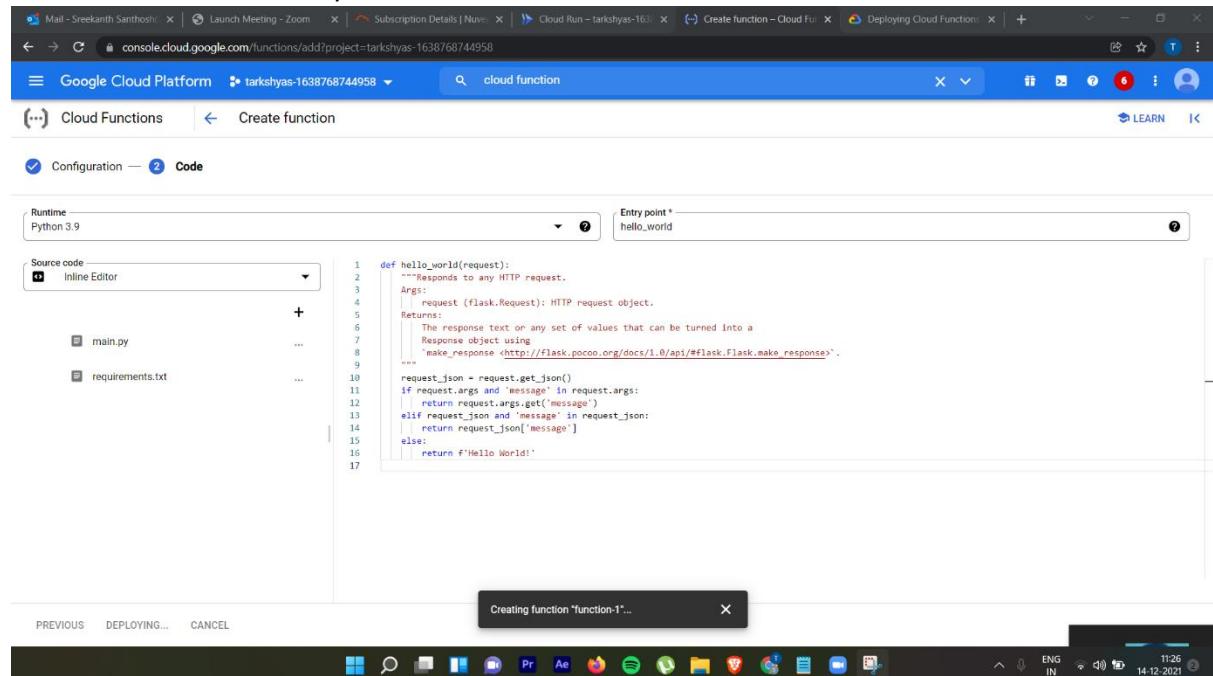
5.Create a cloud function to host a http function

STEP 1: Create a Cloud Function

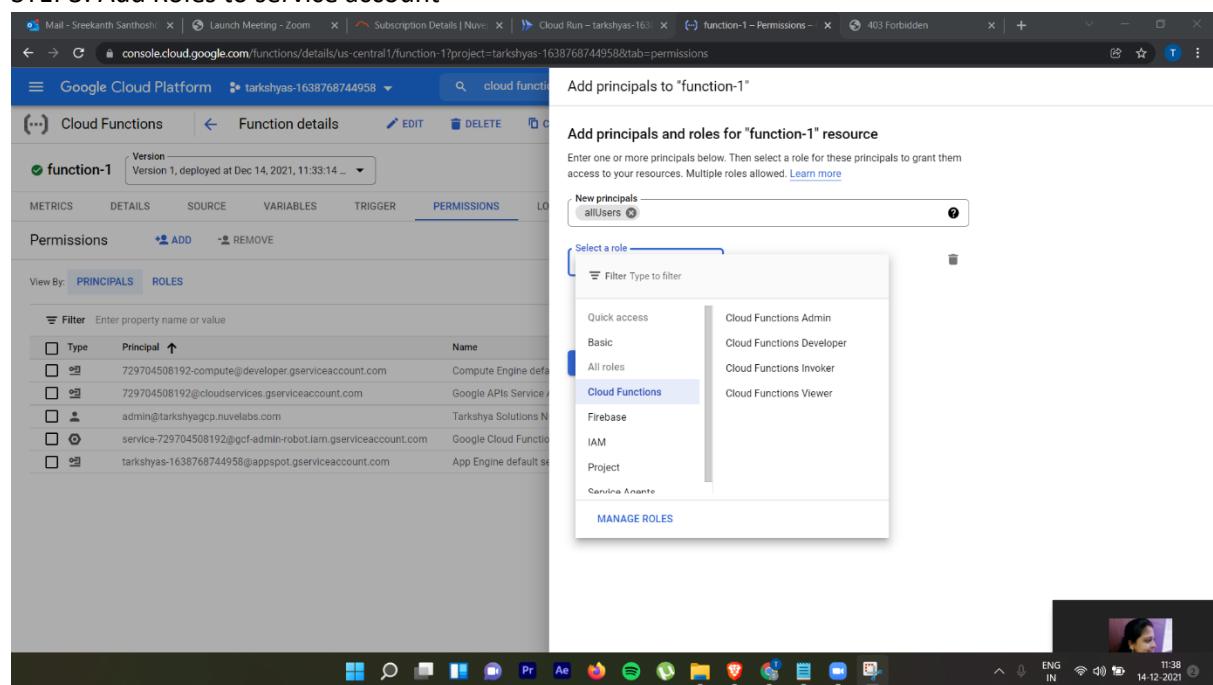
The screenshot shows a browser window with the URL `console.cloud.google.com/functions/add?project=tarkshyas-1638768744958`. The page title is "Create function". The main content area is titled "Cloud Functions" and contains a brief description of what Google Cloud Functions are. Below the description is a "CREATE FUNCTION" button.

The "Create function" wizard has two tabs: "Configuration" (selected) and "Code". The "Configuration" tab is divided into sections: "Basics" (Function name: "function-1", Region: "us-central1"), "Trigger" (HTTP trigger type, URL: `https://us-central1-tarkshyas-1638768744958.cloudfunctions.net/function-1`, Authentication: "Require authentication" selected), and "NEXT" and "CANCEL" buttons at the bottom.

STEP 2: Select runtime as Python 3.9



STEP3: Add Roles to service account



Sreekanth Santhosh-198642

The screenshot shows the Google Cloud Platform interface for managing permissions on a Cloud Function named 'function-1'. The 'PERMISSIONS' tab is selected. A table lists the following principals and their roles:

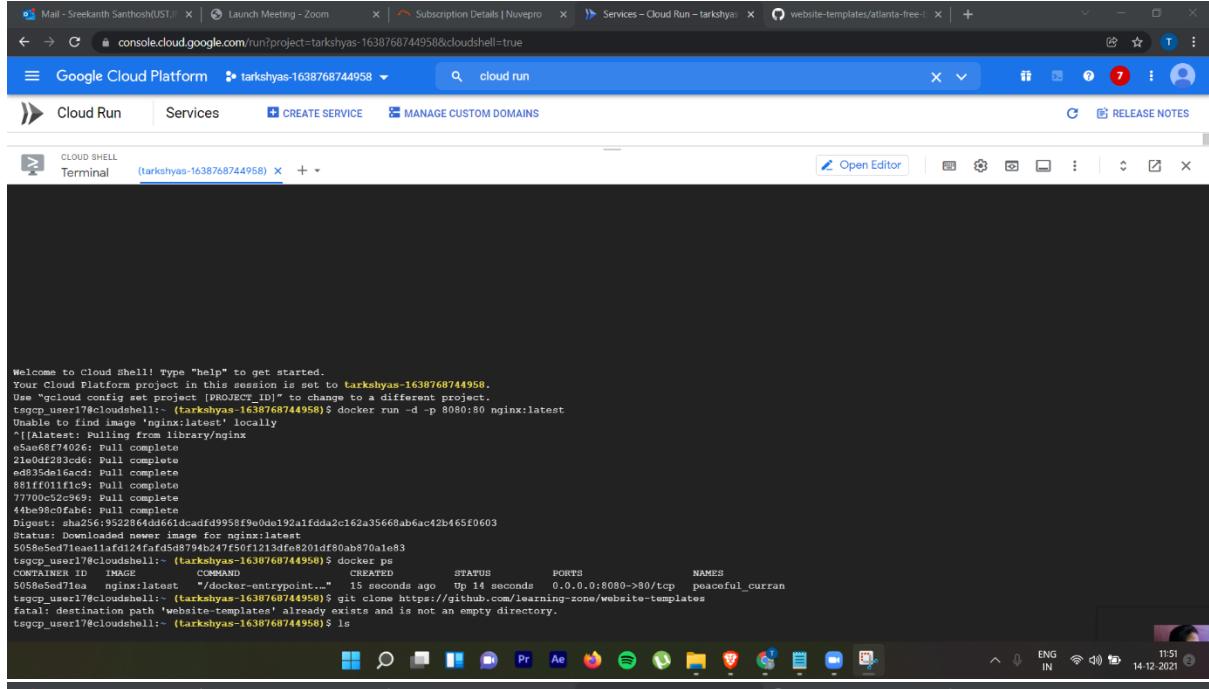
Type	Principal	Name	Role	Inheritance
Compute Engine default service account	729704508192-compute@developer.gserviceaccount.com	Compute Engine default service account	Editor	tarkshyas-1638768744958
Google APIs Service Agent	729704508192@cloudservices.gserviceaccount.com	Google APIs Service Agent	Editor	tarkshyas-1638768744958
Tarkshya Solutions Nuverprotechnologies	admin@tarkshyagcp.nuvelabs.com	Tarkshya Solutions Nuverprotechnologies	Owner	tarkshyas-1638768744958
Cloud Functions Invoker	allUsers	Cloud Functions Service Agent	Cloud Functions Service Agent	tarkshyas-1638768744958
App Engine default service account	service-729704508192@gcf-admin-robot.lam.gserviceaccount.com	App Engine default service account	Editor	tarkshyas-1638768744958
	tarkshyas-1638768744958.appspot.gserviceaccount.com			tarkshyas-1638768744958

The screenshot shows a Windows desktop environment. A browser window is open, displaying the output of the Cloud Function. The URL is `us-central1-tarkshyas-1638768744958.cloudfunctions.net/function-1`. The page content is "Hello World!".

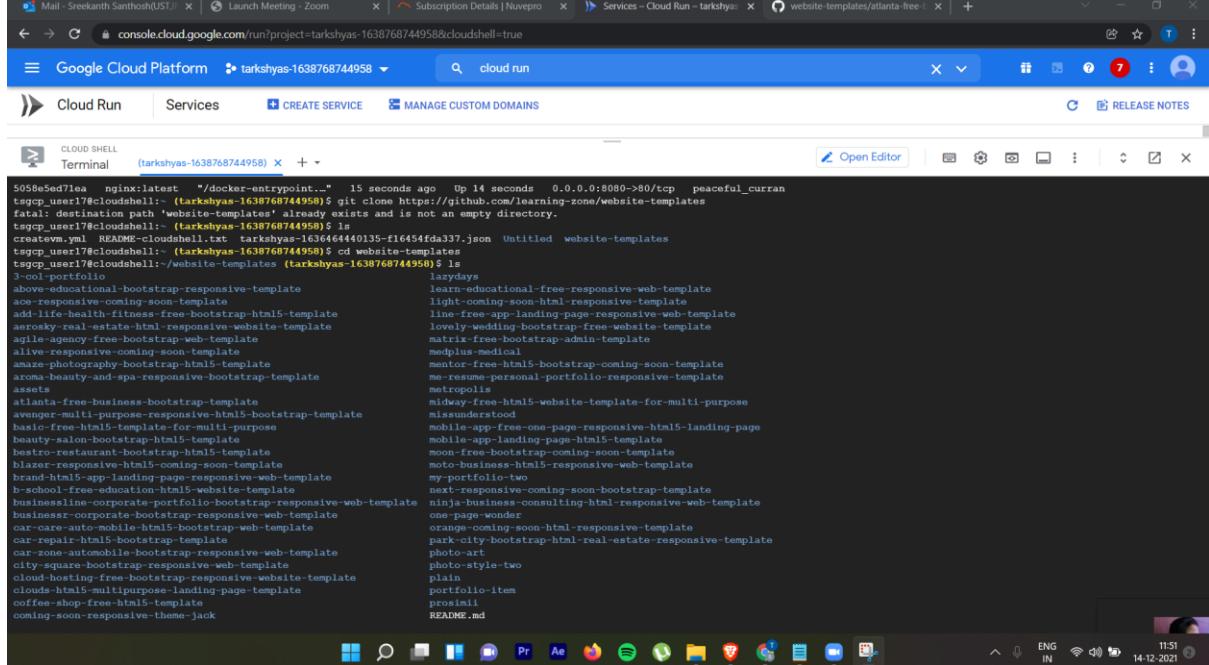
The screenshot shows a Windows desktop environment. A browser window is open, displaying the output of the Cloud Function. The URL is `us-central1-tarkshyas-1638768744958.cloudfunctions.net/function-1`. The page content is "Hello World!".

6.Cloud run

STEP1: Create a Docker Image



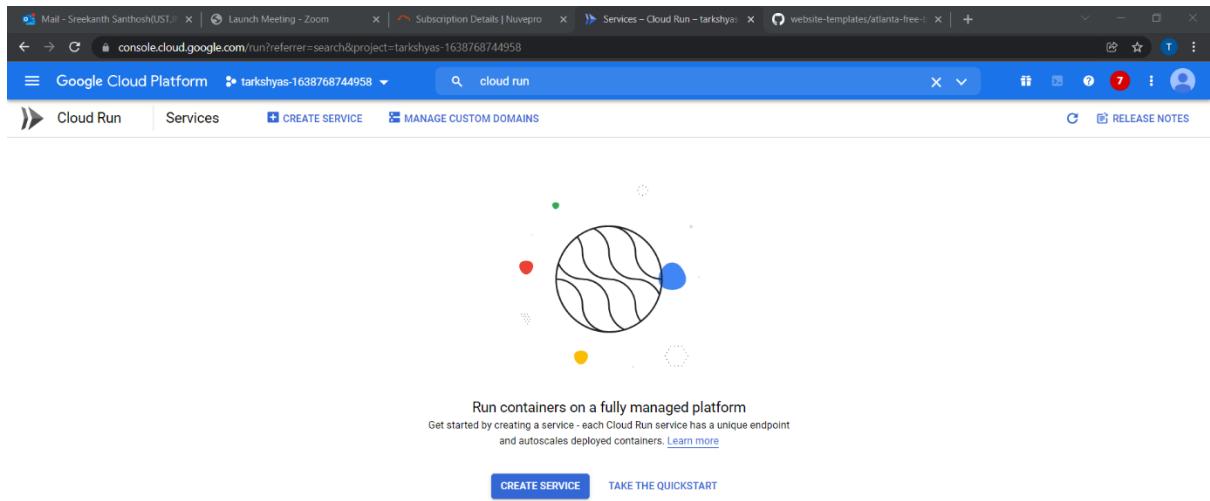
```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to tarkshyas-1638768744958.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
tarkshyas@172-204-111-101: ~ (tarkshyas-1638768744958)$ docker run -d -p 8080:80 nginx:latest
Unable to find image 'nginx:latest' locally
"latest": Pulling from library/nginx
e5a68f74026: Pull complete
21e0df233cd6: Pull complete
ed835de1eacd: Pull complete
881f011f1c9: Pull complete
77700c52d969: Pull complete
44be98c0f016: Pull complete
Digest: sha256:95296464164045d4adfd49958f9e0de192a1fdd2c162a3566ab6ac42b465f0603
Status: Downloaded newer image for nginx:latest
5058e5ed1ea11fd124fd5d08794c247f50f1213fe8e201df80ab870a1e83
tsgcp_user17@cloudshell:~ (tarkshyas-1638768744958)$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
5058e5ed1ea11fd124fd5d08794c247f50f1213fe8e201df80ab870a1e83
tsgcp_user17@cloudshell:~ (tarkshyas-1638768744958)$ git clone https://github.com/learning-zone/website-templates
fatal: destination path 'website-templates' already exists and is not an empty directory.
tsgcp_user17@cloudshell:~ (tarkshyas-1638768744958)$ ls
```



```
5058e5ed1ea11fd124fd5d08794c247f50f1213fe8e201df80ab870a1e83
tsgcp_user17@cloudshell:~ (tarkshyas-1638768744958)$ git clone https://github.com/learning-zone/website-templates
fatal: destination path 'website-templates' already exists and is not an empty directory.
tsgcp_user17@cloudshell:~ (tarkshyas-1638768744958)$ cd website-templates
tsgcp_user17@cloudshell:~/website-templates (tarkshyas-1638768744958)$ ls
3-col-portfolio
above-educational-bootstrap-responsive-template
ace-responsive-coming-soon-template
add-life-health-fitness-free-bootstrap-html5-template
aerovsky-real-estate-html5-responsive-website-template
agile-agency-free-bootstrap-template
alive-responsive-coming-soon-template
amaze-photography-bootstrap-html5-template
aroma-beauty-and-spa-responsive-bootstrap-template
assets
atlanta-free-business-bootstrap-template
avenger-multi-purpose-responsive-html5-bootstrap-template
basic-free-html5-template-for-multi-purpose
beauty-salon-bootstrap-html5-template
best-reviews-bootstrap-template
blazer-responsive-html5-free-template
bootstrap-html5-landing-page-responsive-web-template
b-school-free-education-html5-website-template
businessline-corporate-portfolio-bootstrap-responsive-web-template
business-corporate-bootstrap-responsive-web-template
car-care-auto-mobile-html5-bootstrap-web-template
car-repair-html5-bootstrap-template
car-zone-automobile-bootstrap-responsive-web-template
city-square-bootstrap-responsive-web-template
cloud-music-free-bootstrap-responsive-website-template
clouds-html5-attic-free-landing-page-template
coffee-shop-free-html5-template
coming-soon-responsive-theme-jack
README.md
```

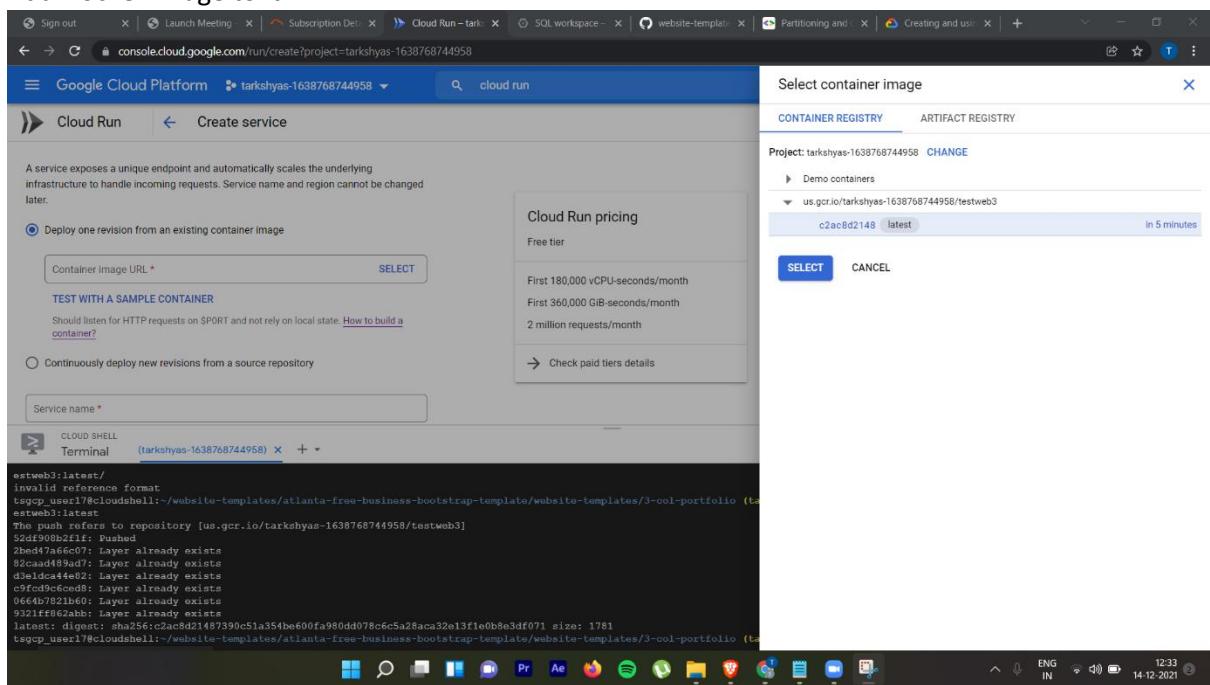
Sreekanth Santhosh-198642

STEP 2: Create a Cloud Run Service

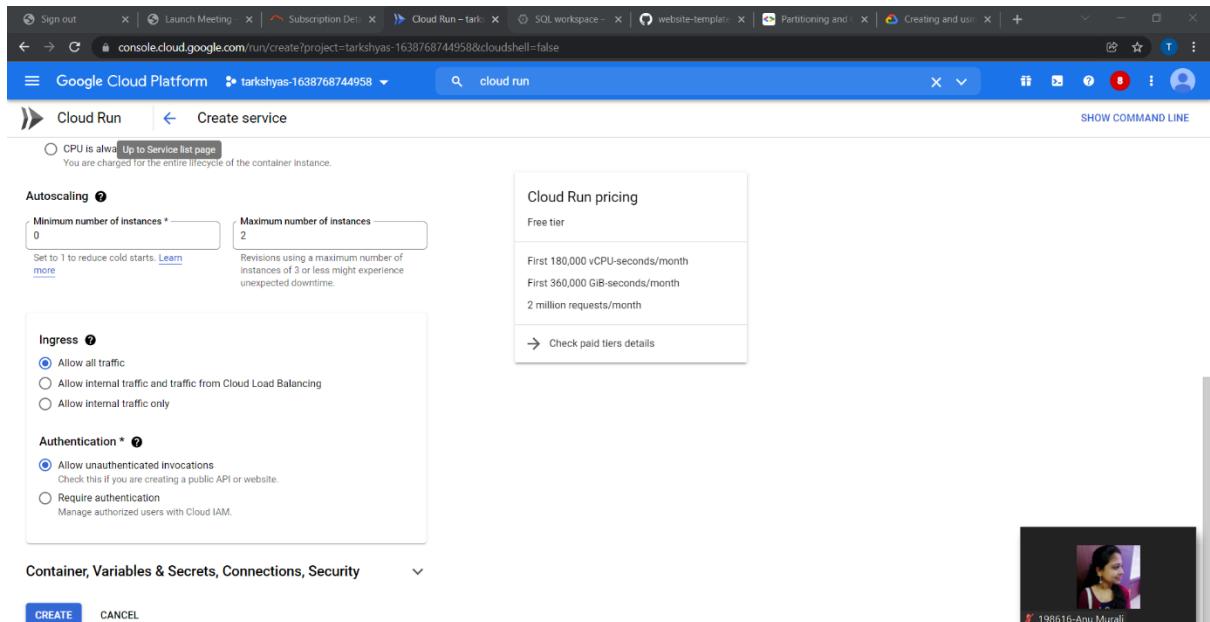


S

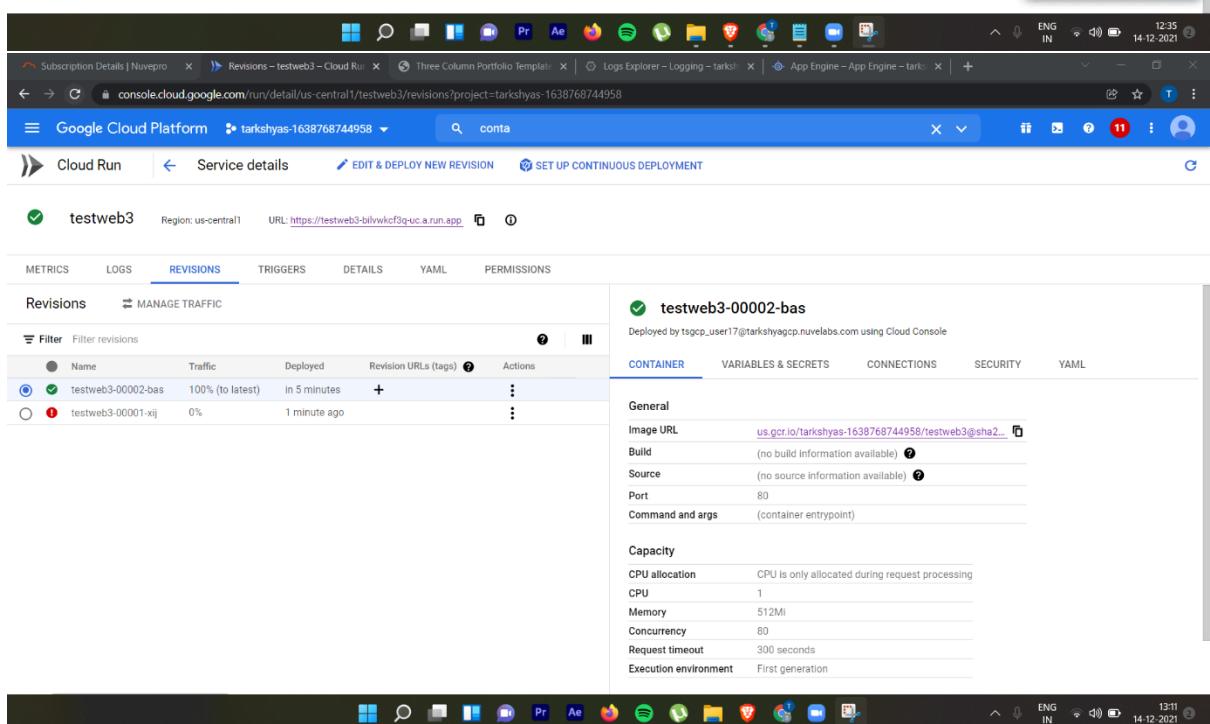
Add Docker Image to it



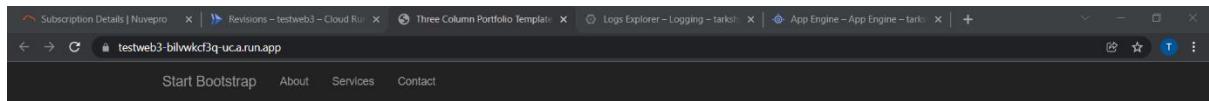
Sreekanth Santhosh-198642



The screenshot shows the Google Cloud Platform Cloud Run creation interface. It includes fields for Autoscaling (Minimum number of instances: 0, Maximum number of instances: 2), Ingress (Allow all traffic selected), Authentication (Allow unauthenticated invocations selected), and Cloud Run pricing (Free tier details). A sidebar on the right shows a profile picture of Anu Murali.



The screenshot shows the Google Cloud Platform Cloud Run service details page for 'testweb3'. It displays two revisions: 'testweb3-00002-bas' (selected) and 'testweb3-00001-xij'. The 'REVISIONS' tab is active. The right panel shows the container configuration for 'testweb3-00002-bas', including General settings (Image URL: us.gcr.io/tarkshyas-1638768744958/testweb3@sha2_...), Capacity (CPU allocation: 1, Memory: 512Mi), and Environment variables (PORT: 80).



3 Col Portfolio Showcase Your Work



Project Name

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nam viverra euismod odio, gravida pellentesque urna
varius vitae.



Project Name

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nam viverra euismod odio, gravida pellentesque urna
varius vitae.



Project Name

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nam viverra euismod odio, gravida pellentesque urna
varius vitae.

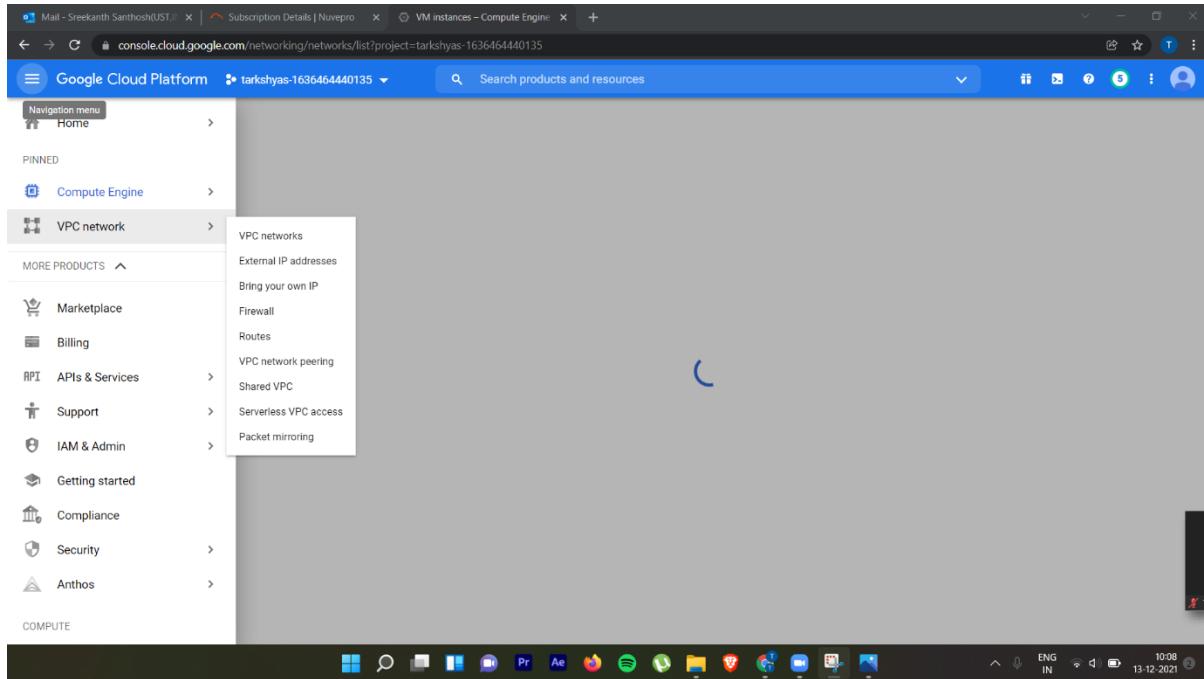
« 1 2 3 4 5 »

Copyright © Company 2021



7.VPC peering

STEP 1: Create a VPC Network



Create 1 VPC with 2 sub-nets(to show 2 VMs with same VPC Network can communicate)

The screenshot shows the 'VPC network details' page for 'vpc1'. The left sidebar lists various VPC-related options. The main area displays the subnet configuration for 'vpc1'. It shows two subnets: 'europe-west1' (Region: europe-west1, IP address ranges: 10.132.0.0/20, Gateway: 10.132.0.1, Private Google Access: Off, Flow logs: Off) and 'us-central1' (Region: us-central1, IP address ranges: 10.0.0.0/9, Gateway: 10.0.0.1, Private Google Access: Off, Flow logs: Off). Below this, a section for 'Reserved subnets for internal HTTP(S) load balancers' is shown, which is currently empty.

Name	Region	IP address ranges	Gateway	Private Google Access	Flow logs
europe-west1	europe-west1	10.132.0.0/20	10.132.0.1	Off	Off
us-central1	us-central1	10.0.0.0/9	10.0.0.1	Off	Off

The screenshot shows the 'VPC networks' list page. The left sidebar is identical to the previous screen. The main area lists various VPC networks, including 'northamerica-northeast1', 'europe-west4', 'europe-north1', 'us-west2', 'asia-east2', 'europe-west5', 'asia-northeast2', 'asia-northeast3', 'us-west3', 'us-west4', 'asia-southeast2', 'europe-central2', 'northamerica-northeast2', 'asia-south2', 'australia-southeast2', 'southamerica-west1', and 'vpc1'. The 'vpc1' entry is expanded, showing its details: Region: 2, IP range: 1460, Subnet mode: Custom, and two subnets: 'us-central1' (IP range: 10.0.0.0/9, Gateway: 10.0.0.1, Private Google Access: Off, Flow logs: Off) and 'europe-west1' (IP range: 10.132.0.0/20, Gateway: 10.132.0.1, Private Google Access: Off, Flow logs: Off).

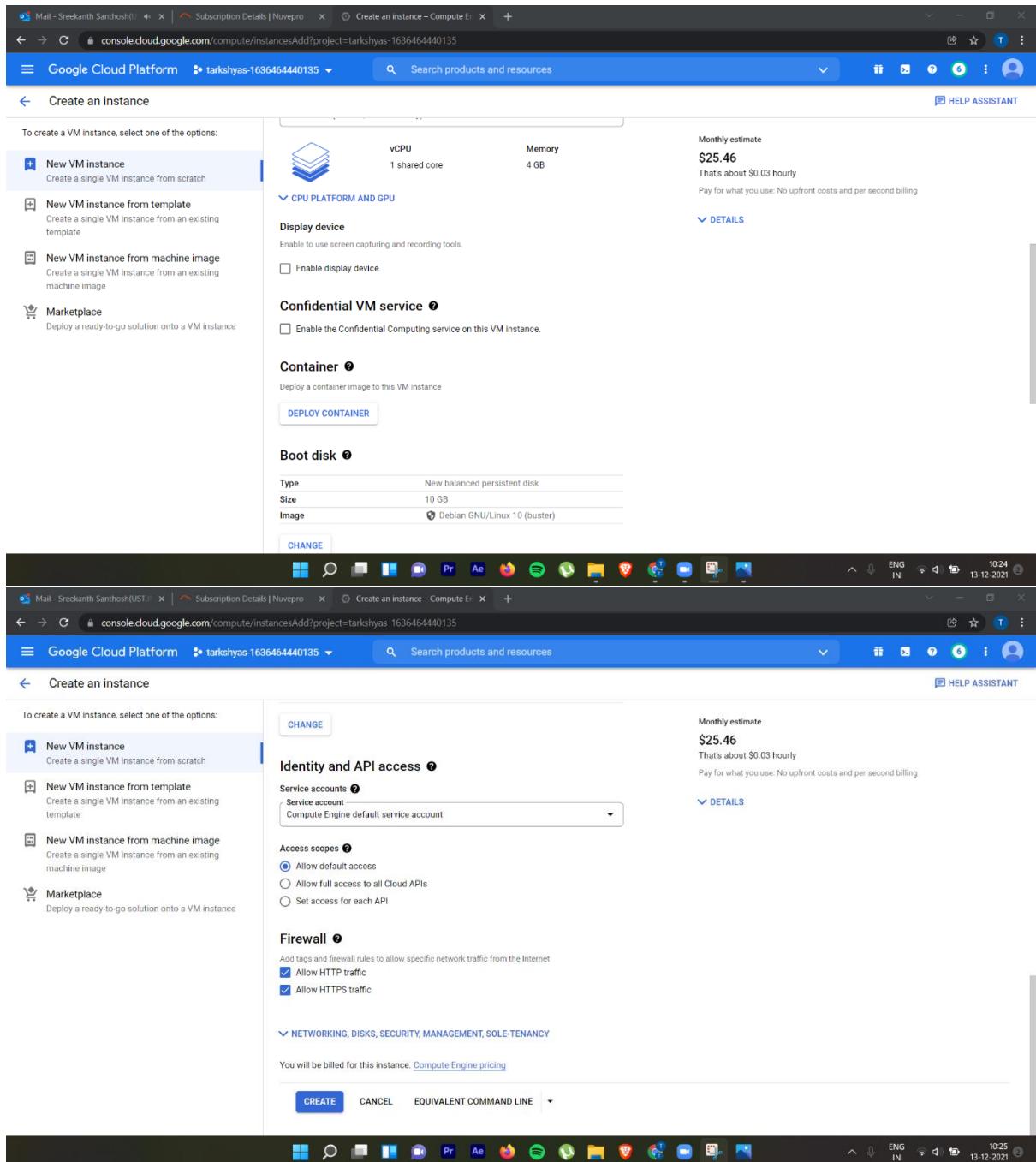
Name	Region	IP address ranges	Gateway	Private Google Access	Flow logs
us-central1	us-central1	10.0.0.0/9	10.0.0.1	Off	Off
europe-west1	europe-west1	10.132.0.0/20	10.132.0.1	Off	Off

STEP 2: Create 2 VM instances

The screenshot shows the 'Create an instance' page in the Google Cloud Platform Compute Engine interface. The configuration details are as follows:

- Name:** instance-1
- Region:** us-central1 (Iowa)
- Zone:** us-central1-a
- Machine Configuration:**
 - Machine family:** GENERAL-PURPOSE
 - Series:** E2
 - Machine type:** e2-medium (2 vCPU, 4 GB memory)
 - vCPU:** 2
 - Memory:** 4 GB
- Billing:** Monthly estimate: \$25.46 (That's about \$0.03 hourly). Pay for what you use: No upfront costs and per second billing.

Sreekanth Santhosh-198642



The screenshot shows the Google Cloud Platform Compute Engine interface for creating a new VM instance. The left sidebar lists options: New VM instance (selected), New VM instance from template, New VM instance from machine image, and Marketplace. The main panel shows configuration details:

- vCPU:** 1 shared core
- Memory:** 4 GB
- Monthly estimate:** \$25.46
- Display device:** Enabled
- Confidential VM service:** Enabled
- Container:** Deploy a container image to this VM instance
- Boot disk:** Type: New balanced persistent disk, Size: 10 GB, Image: Debian GNU/Linux 10 (buster)

Below these settings are sections for Identity and API access (Service account: Compute Engine default service account, Access scopes: Allow default access selected), Firewall (Allow HTTP traffic, Allow HTTPS traffic selected), and Networking, Disks, Security, Management, Sole-Tenancy.

Connect the same Network created before

The screenshot shows the 'Create an instance' page in the Google Cloud Platform Compute Engine interface. On the left, there's a sidebar with options like 'New VM instance', 'New VM instance from template', 'New VM instance from machine image', and 'Marketplace'. The main area is titled 'Set a custom hostname for this instance or leave it default. Choice is permanent' and includes an 'IP forwarding' section with a checkbox for 'Enable'. Below this is the 'Network interfaces' section, which lists 'default default (10.128.0.0/20)' and allows for adding new interfaces. The 'New network interface' form shows fields for 'Network' (set to 'vpc1'), 'Subnetwork' (set to 'us-central-1 (10.0.0.0/9)'), and 'Primary internal IP' (set to 'Ephemeral (Automatic)'). There are also sections for 'Alias IP ranges' and 'External IP' (set to 'Ephemeral'). To the right, a 'Monthly estimate' table shows a cost of '\$25.46' for about \$0.03 hourly, noting no upfront costs and per second billing.

Create similar instance

The screenshot shows the 'VM instances' page in the Google Cloud Platform Compute Engine interface. The left sidebar has sections for 'Virtual machines' (with 'VM Instances' selected), 'Storage' (with 'Disks', 'Snapshots', 'Images'), and 'Marketplace'. The main area is titled 'INSTANCES INSTANCE SCHEDULE' and displays a table of existing VM instances. The table columns are: Status, Name, Zone, Recommendations, In use by, Internal IP, External IP, and Connect. Two instances are listed: 'instance-1' in 'us-central1-a' with IP 10.0.0.2 (nic0) and 'instance-2' in 'europe-west1-b' with IP 10.132.0.2 (nic0). Both instances have an 'SSH' button in the 'Connect' column.

STEP 3 : Add Firewall Rule to VPC Network

The screenshot shows two browser windows side-by-side.

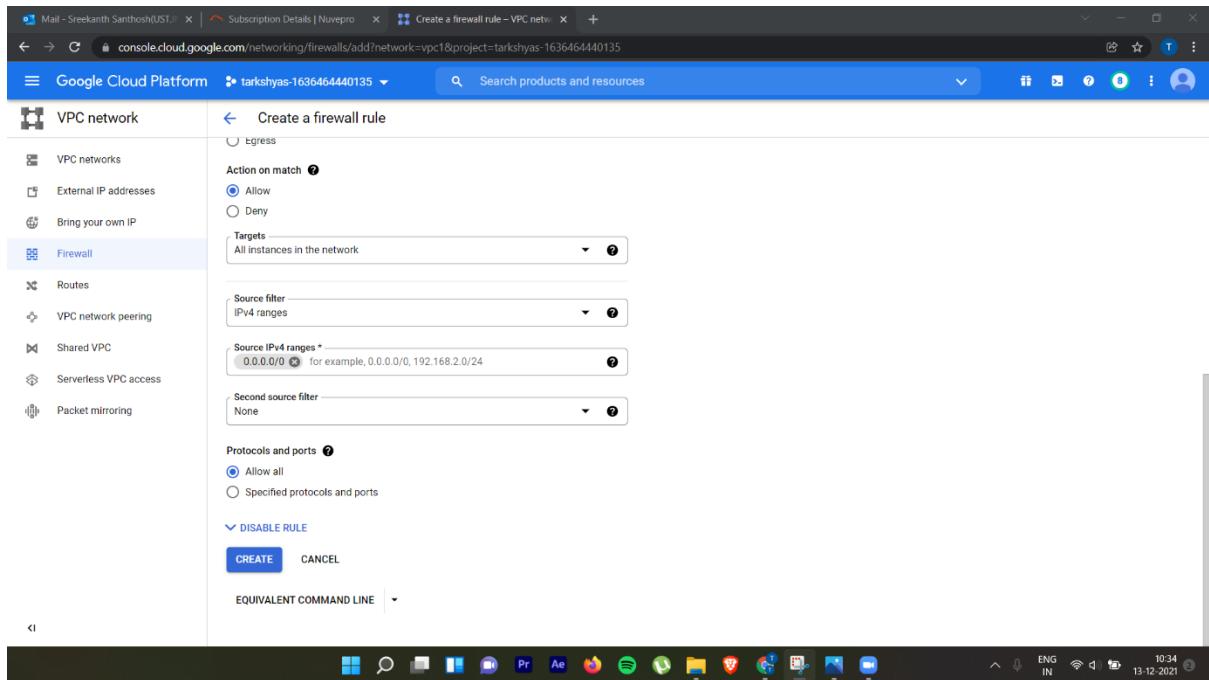
Left Window (VPC Network Details):

- URL: `console.cloud.google.com/networking/networks/details/vpc1?project=tarkshyas-163646440135&pageTab=FIREWALLS`
- Panel: **VPC network** (selected)
- Sub-panel: **Firewall**
- Content: Shows existing firewall rules:

Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Logs	Hit count	Last hit	Insights
vpc1-allow-https	Ingress	https-server	IP ranges	tcp:443	Allow	1000	Off	—	—	▼
vpc1-allow-httpp	Ingress	http-server	IP ranges	tcp:80	Allow	1000	Off	—	—	▼

Right Window (Create a firewall rule):

- URL: `console.cloud.google.com/networking/firewalls/add?network=vpc1&project=tarkshyas-163646440135`
- Panel: **VPC network** (selected)
- Sub-panel: **Firewall**
- Content: Form fields for creating a new rule:
 - Name ***: `firewall-rule1`
 - Description**: (empty)
 - Logs**: Turning on firewall logs can generate a large number of logs which can increase costs in Cloud Logging. Learn more
 - On
 - Off
 - Network ***: `vpc1`
 - Priority ***: `1000` (checkbox: `CHECK PRIORITY OF OTHER FIREWALL RULES`)
 - Direction of traffic**:
 - Ingress
 - Egress
 - Action on match**:
 - Allow
 - Deny



We can See 2 VMs under Same VPC Network Communicating.But this doesn't work for 2 VMs with 2 different VPC Network.So we do VPC Peering.

```
tsgcp_user17@instance-1: ~ - Google Chrome
ssh.cloud.google.com/projects/tarkshyas-1636464440135/zones/us-central1-a/instances/instance-1?authuser=0
Connected, host fingerprint: ssh-rsa 0 80:E2:5C:2B:D1:88:51:18:CAB3:EA:42:03
Linux instance-1 4.19.0-18-cloud-amd64 #1 SMP Debian 4.19.208-1 (2021-09-29) 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.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
tsgcp_user17@instance-1:~$ ping 10.132.0.2
PING 10.132.0.2 (10.132.0.2) 56(84) bytes of data.
64 bytes from 10.132.0.2: icmp_seq=1 ttl=64 time=100 ms
64 bytes from 10.132.0.2: icmp_seq=2 ttl=64 time=98.9 ms
64 bytes from 10.132.0.2: icmp_seq=3 ttl=64 time=98.9 ms
64 bytes from 10.132.0.2: icmp_seq=4 ttl=64 time=98.9 ms
64 bytes from 10.132.0.2: icmp_seq=5 ttl=64 time=98.9 ms
64 bytes from 10.132.0.2: icmp_seq=6 ttl=64 time=98.9 ms
64 bytes from 10.132.0.2: icmp_seq=7 ttl=64 time=98.9 ms
64 bytes from 10.132.0.2: icmp_seq=8 ttl=64 time=98.9 ms
64 bytes from 10.132.0.2: icmp_seq=9 ttl=64 time=98.9 ms
64 bytes from 10.132.0.2: icmp_seq=10 ttl=64 time=98.9 ms
64 bytes from 10.132.0.2: icmp_seq=11 ttl=64 time=98.9 ms
64 bytes from 10.132.0.2: icmp_seq=12 ttl=64 time=98.9 ms
64 bytes from 10.132.0.2: icmp_seq=13 ttl=64 time=98.9 ms
64 bytes from 10.132.0.2: icmp_seq=14 ttl=64 time=98.9 ms
64 bytes from 10.132.0.2: icmp_seq=15 ttl=64 time=98.9 ms
64 bytes from 10.132.0.2: icmp_seq=16 ttl=64 time=98.9 ms
64 bytes from 10.132.0.2: icmp_seq=17 ttl=64 time=98.9 ms
64 bytes from 10.132.0.2: icmp_seq=18 ttl=64 time=98.9 ms
64 bytes from 10.132.0.2: icmp_seq=19 ttl=64 time=98.9 ms
tsgcp_user17@instance-2: ~ - Google Chrome
ssh.cloud.google.com/projects/tarkshyas-1636464440135/zones/europe-west1-b/instances/instance-2?authuser=0
Connected, host fingerprint: ssh-rsa 0:CD:9F:24:F7:16:33:2F:7B:D7:D1:1C:18
Linux instance-2 4.19.0-18-cloud-amd64 #1 SMP Debian 4.19.208-1 (2021-09-29) 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.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
tsgcp_user17@instance-2:~$ ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=98.9 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=98.9 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=98.9 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=98.9 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=98.9 ms
```

STEP 4: Create VPC2

The screenshot shows the Google Cloud Platform interface for creating a VPC network. 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 a form for creating a new VPC. The 'Name' field is set to 'vpc2'. Below it is a 'Description' field. A section titled 'Subnets' explains that subnets let you create your own private cloud topology within Google Cloud. It offers two creation modes: 'Automatic' (selected) and 'Custom'. A 'New subnet' dialog box is open, showing fields for 'Name' (set to 'subnet1') and 'Description'. The bottom of the screen shows a taskbar with various application icons and system status indicators.

Sreekanth Santhosh-198642

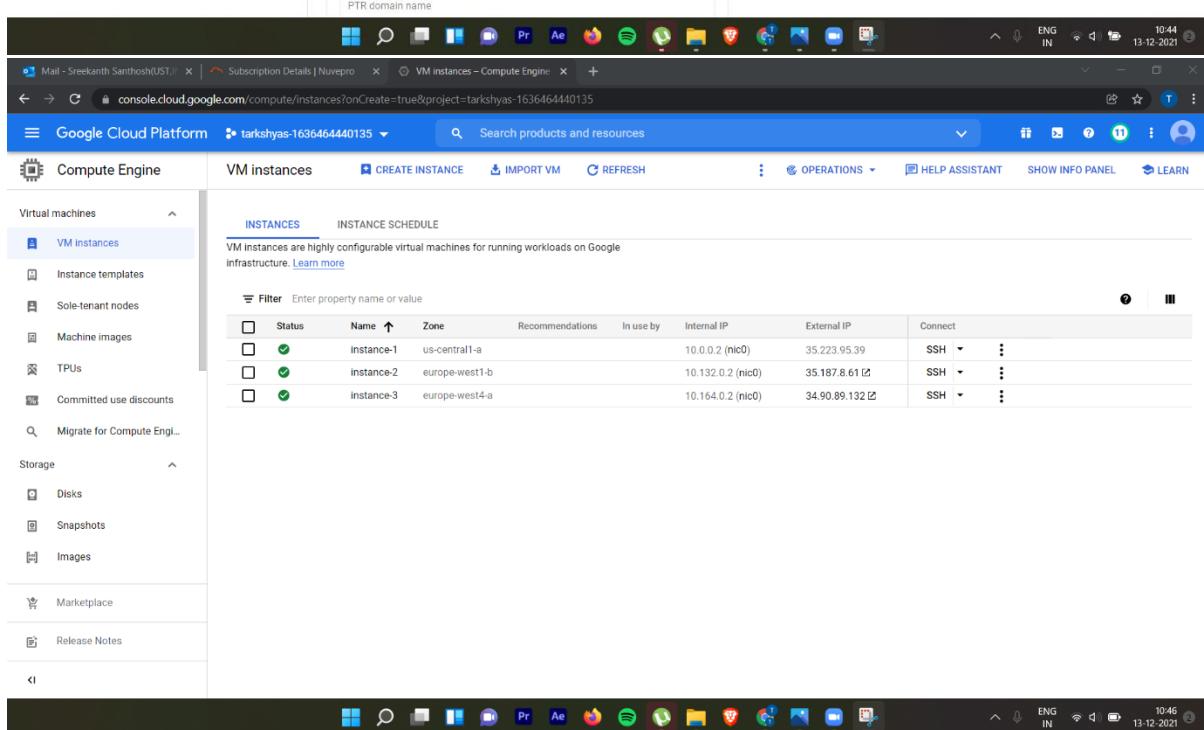
The screenshot shows two consecutive screenshots of the Google Cloud Platform interface.

Top Screenshot: A 'Create a VPC network' dialog box. The 'Name' field is set to 'europe-west4'. The 'Region' field is set to 'europe-west4'. The 'IP address range' field is set to '10.164.0.0/20'. Under 'Private Google Access', the 'Off' radio button is selected. Under 'Flow logs', the 'Off' radio button is selected. A 'DONE' button is at the bottom right.

Bottom Screenshot: A list of existing VPC networks. The table includes columns for Name, Region, IP Range, and Private Google Access status (Off). The list includes networks like 'europe-north1', 'us-west2', 'asia-east2', etc., and two custom networks named 'vpc1' and 'vpc2'.

Name	Region	IP Range	Private Google Access
europe-north1	default	10.166.0.0/20	Off
us-west2	default	10.168.0.0/20	Off
asia-east2	default	10.170.0.0/20	Off
europe-west5	default	10.172.0.0/20	Off
asia-northeast2	default	10.174.0.0/20	Off
asia-northeast3	default	10.178.0.0/20	Off
us-west3	default	10.180.0.0/20	Off
us-west4	default	10.182.0.0/20	Off
asia-southeast2	default	10.184.0.0/20	Off
europe-central2	default	10.186.0.0/20	Off
northamerica-northeast2	default	10.188.0.0/20	Off
asia-south2	default	10.190.0.0/20	Off
australia-southeast2	default	10.192.0.0/20	Off
southamerica-west1	default	10.194.0.0/20	Off
vpc1	2 us-central1 europe-west1	1460 Custom 10.0.0.0/9 10.132.0.0/20	3 Off Off
vpc2	1 europe-west4	1460 Custom 10.164.0.0/20	0 Off Off

STEP 5: Create another instance connected to vpc2



The screenshot shows the Google Cloud Platform Compute Engine VM Instances page. The left sidebar is collapsed, and the main area displays the 'VM instances' tab. The 'INSTANCES' section lists three virtual machines:

Status	Name	Zone	Internal IP	External IP	Connect
✓	Instance-1	us-central1-a	10.0.0.2 (nic0)	35.223.95.39	SSH
✓	instance-2	europe-west1-b	10.132.0.2 (nic0)	35.187.8.61	SSH
✓	instance-3	europe-west4-a	10.164.0.2 (nic0)	34.90.89.132	SSH

STEP 6: Open VPC network Peering

The screenshot shows two instances of the Google Cloud Platform (GCP) web interface. The top instance displays the 'VPC network peering' page under the 'VPC network' section of the sidebar. It features a summary box titled 'VPC network peering' with a brief description of what it does and a 'CREATE CONNECTION' button. The bottom instance shows the 'Create peering connection' form. In this form, the 'Name' field is set to 'vpc1-to-vpc2'. The 'Your VPC network' dropdown is set to 'vpc1'. Under 'Peered VPC network', the 'In project' radio button is selected, and the 'VPC network name' dropdown is set to 'vpc2'. In the 'Exchange custom routes' section, the 'Export custom routes' checkbox is checked. The 'Exchange subnet routes with public IP' section has the 'Export subnet routes with public IP' checkbox checked. At the bottom of the form are 'CREATE' and 'CANCEL' buttons.

Connect vpc1 to vpc2

The screenshot shows the Google Cloud Platform VPC network peering interface. On the left, there's a sidebar with options like VPC networks, External IP addresses, Bring your own IP, Firewall, Routes, VPC network peering (which is selected), Shared VPC, Serverless VPC access, and Packet mirroring. The main area is titled "VPC network peering" and shows a table of existing peering connections. A new connection is being created, indicated by a "CREATE PEERING CONNECTION" button. The table has columns for Name, Your VPC network, Peered VPC network, Peered project ID, Status, Exchange custom routes, and Exchange subnet routes with public IP. One row is selected, showing "vpc1-to-vpc2" as the name, "vpc1" as the your VPC network, "vpc2" as the peered VPC network, "tarkshyas-1636464440135" as the peered project ID, and "Inactive" as the status. There are also "Export custom routes" and "Export subnet routes with public IP" checkboxes. A success message at the bottom says "Successfully created peering connection 'vpc1-to-vpc2'." The system tray at the bottom right shows the date and time as 13-12-2021 11:02.

Connect vpc2 to vpc1

The screenshot shows the "Create peering connection" interface. The sidebar on the left is identical to the previous screenshot. The main form is titled "Create peering connection". It contains several sections: 1) A note stating "Your VPC network will be fully connected to the peered VPC network (full mesh topology). Routes to subnets in the peered VPC network will be automatically created." 2) A "Name" field with "vpc2-to-vpc1" entered, with a note below saying "Lowercase letters, numbers, hyphens allowed." 3) A "Your VPC network" dropdown set to "vpc2". 4) A "Peered VPC network" section with two radio button options: "In project tarkshyas-1636464440135" (selected) and "In another project". 5) A "VPC network name" dropdown set to "vpc1". 6) An "Exchange custom routes" section with a note about importing or exporting static and dynamic routes, and checkboxes for "Import custom routes" (unchecked) and "Export custom routes" (checked). 7) An "Exchange subnet routes with public IP" section with a note about importing or exporting subnet routes with public IP, and checkboxes for "Import subnet routes with public IP" (unchecked) and "Export subnet routes with public IP" (checked). The system tray at the bottom right shows the date and time as 13-12-2021 11:02.

Sreekanth Santhosh-198642

Mail - Sreekanth Santhosh(UST) | Subscription Details | Nuvepro | VPC network peering - VPC networks | console.cloud.google.com/networking/peering/list?project=tarkshyas-1636464440135

Google Cloud Platform tarkshyas-1636464440135 Search products and resources

VPC network VPC network peering + CREATE PEERING CONNECTION C REFRESH ⌂ DELETE

Filter Enter property name or value

Name	Your VPC network	Peered VPC network	Peered project ID	Status	Exchange custom routes	Exchange subnet routes with public IP
vpc1-to-vpc2	vpc1	vpc2	tarkshyas-1636464440135	Active	Export custom routes	Export subnet routes with public IP
vpc2-to-vpc1	vpc2	vpc1	tarkshyas-1636464440135	Active	Export custom routes	Export subnet routes with public IP

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

Successfully created peering connection "vpc2-to-vpc1".

Successfully VPC Peering is done

ssh.cloud.google.com/projects/tarkshyas-1636464440135/zones/us-central1-a/instances/instance-1~authuser=0... Connected, host fingerprint: ssh-rsa 0 BD:E4:C5:81:26:33:9B:0A:2B:67:44:9B:52:9A :CD:90:0D:24:05:FF:3C:68:11:F9:05:04:2C:22:32:A9:65:07 Linux instance-1 4.19.0-18-cloud-amd64 #1 SMP Debian 4.19.208-1 (2021-09-29) 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.

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

Last login: Mon Dec 13 05:25:59 2021 from 35.235.244.33

root@instance-1:~\$ ping 10.164.0.2

PING 10.164.0.2 (10.164.0.2) 56(84) bytes of data.

64 bytes from 10.164.0.2: icmp_seq=1 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=2 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=3 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=4 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=5 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=6 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=7 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=8 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=9 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=10 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=11 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=12 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=13 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=14 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=15 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=16 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=17 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=18 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=19 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=20 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=21 ttl=64 time=102 ms

64 bytes from 10.164.0.2: icmp_seq=22 ttl=64 time=102 ms

ssh.cloud.google.com/projects/tarkshyas-1636464440135/zones/europe-west4-a/instances/instance-3~authuser=0... Connected, host fingerprint: ssh-rsa 0 BD:E4:C5:81:26:33:9B:0A:2B:67:44:9B:52:9A :3E:90:62:3A:8A:6A:20:3E:13:77:29:37:57:DA:AO:16:CB:8F Linux instance-3 4.19.0-18-cloud-amd64 #1 SMP Debian 4.19.208-1 (2021-09-29) 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.

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

Last login: Mon Dec 13 05:55:58 2021 from 35.235.244.17

root@instance-3:~\$ ping 10.0.0.2

PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.

64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=102 ms

64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=102 ms

64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=102 ms

8.Cloud DNS

STEP 1: Create a Domain

The screenshot shows the Freenom Client Area interface. At the top, there are several tabs: 'Mail - Sreekanth Santhosh(UST)', 'Launch Meeting - Zoom', 'Subscription Details | Nuvapro', 'Client Area - Freenom', and a '+' button. Below the tabs, the Freenom logo and slogan 'A Name for Everyone' are displayed. The main content area is titled 'My Domains' with the sub-instruction 'View & manage all the domains you have registered with us from here...'. A search bar 'Enter Domain to Find' and a blue 'Filter' button are located above a table. The table has columns: 'Domain', 'Registration Date', 'Expiry date', 'Status', and 'Type'. One row is visible, showing 'srk10.gq' (Registration Date: 2021-11-19, Expiry Date: 2022-11-19, Status: ACTIVE, Type: Free), with a 'Manage Domain' button. Below the table, it says 'Results Per Page: 10' and '1 Records Found, Page 1 of 1'. The bottom of the screen shows a Windows taskbar with various icons and system status.

STEP 2: Create a VM instance

The screenshot shows the Google Cloud Platform Compute Engine VM instances page. The URL in the address bar is 'console.cloud.google.com/compute/instances?project=tarkshyas-1638768737406'. The page header includes the GCP logo, project name 'tarkshyas-1638768737406', a search bar 'Search products and resources', and navigation links like 'CREATE INSTANCE', 'REFRESH', 'START / RESUME', 'OPERATIONS', 'HELP ASSISTANT', and 'SHOW INFO PANEL'. On the left, a sidebar menu for 'Compute Engine' is open, showing 'Virtual machines' expanded, with 'VM Instances' selected. Other options include 'Instance templates', 'Sole-tenant nodes', 'Machine images', 'TPUs', 'Committed use discounts', and 'Migrate for Compute Eng...'. Below the sidebar, a 'VM Instances' section features a circular diagram with colored dots (green, red, blue, yellow) representing different instance states. A descriptive text block explains Compute Engine and provides 'CREATE INSTANCE' and 'TAKE THE QUICKSTART' buttons. The bottom of the screen shows a Windows taskbar.

Create an instance

To create a VM instance, select one of the options:

- New VM instance** Create a single VM instance from scratch
- New VM instance from template** Create a single VM instance from an existing template
- New VM instance from machine image** Create a single VM instance from an existing machine image
- Marketplace** Deploy a ready-to-go solution onto a VM instance

Name * instance-1

Labels [+ ADD LABELS](#)

Region * asia-south1 (Mumbai) **Zone *** asia-south1-c

Machine configuration

Machine family GENERAL-PURPOSE COMPUTE-OPTIMIZED MEMORY-OPTIMIZED

Series E2

Machine type e2-medium (2 vCPU, 4 GB memory)

vCPU	Memory
1 shared core	4 GB

CPU PLATFORM AND GPU

Edit network interface

Network * default

Subnetwork * default (10.160.0.0/20)

Primary internal IP Ephemeral (Automatic)

Alias IP ranges [+ ADD IP RANGE](#)

External IP

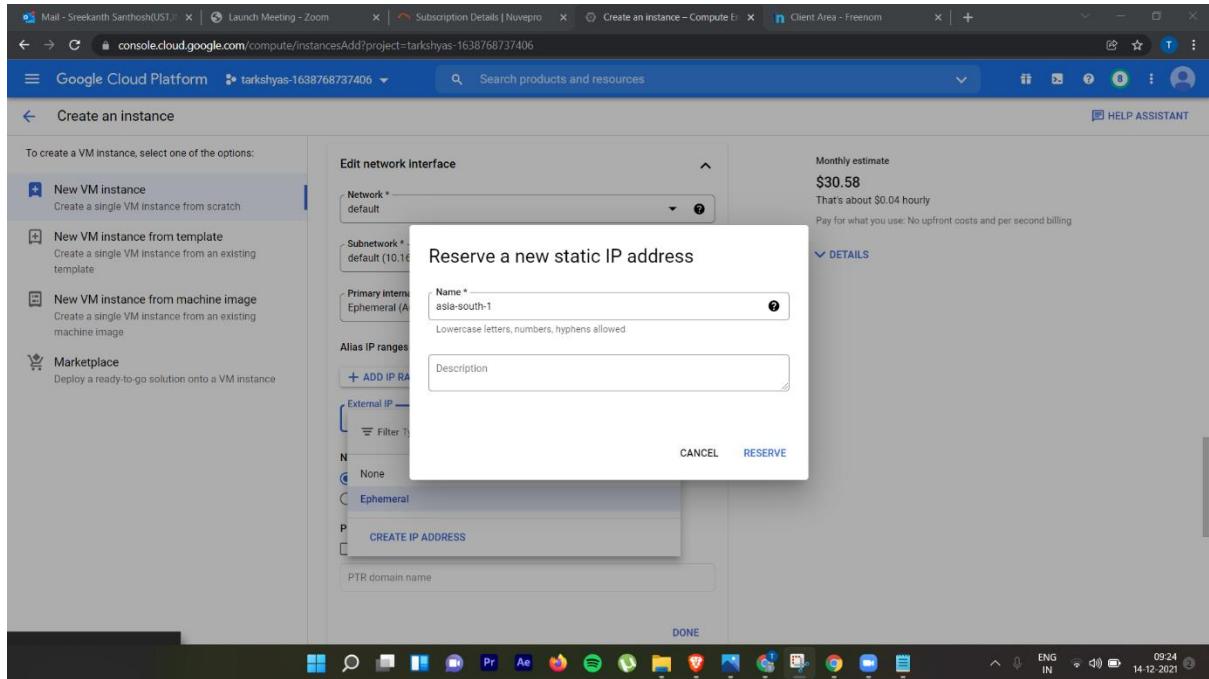
- None
- Ephemeral
- CREATE IP ADDRESS**

Monthly estimate \$30.58

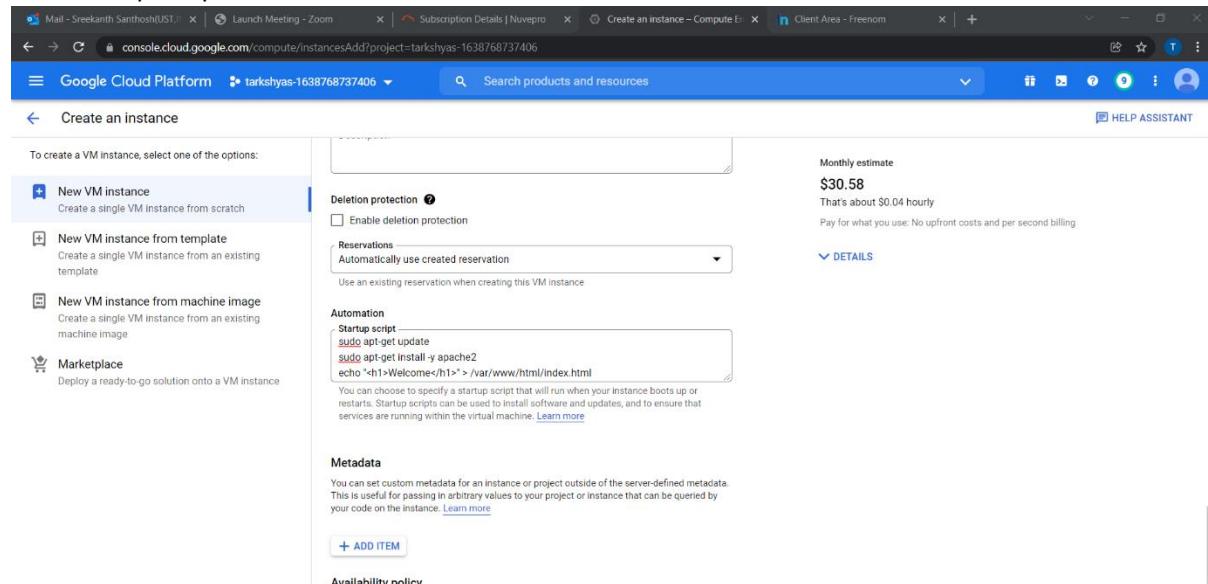
That's about \$0.04 hourly
Pay for what you use: No upfront costs and per second billing

DETAILS

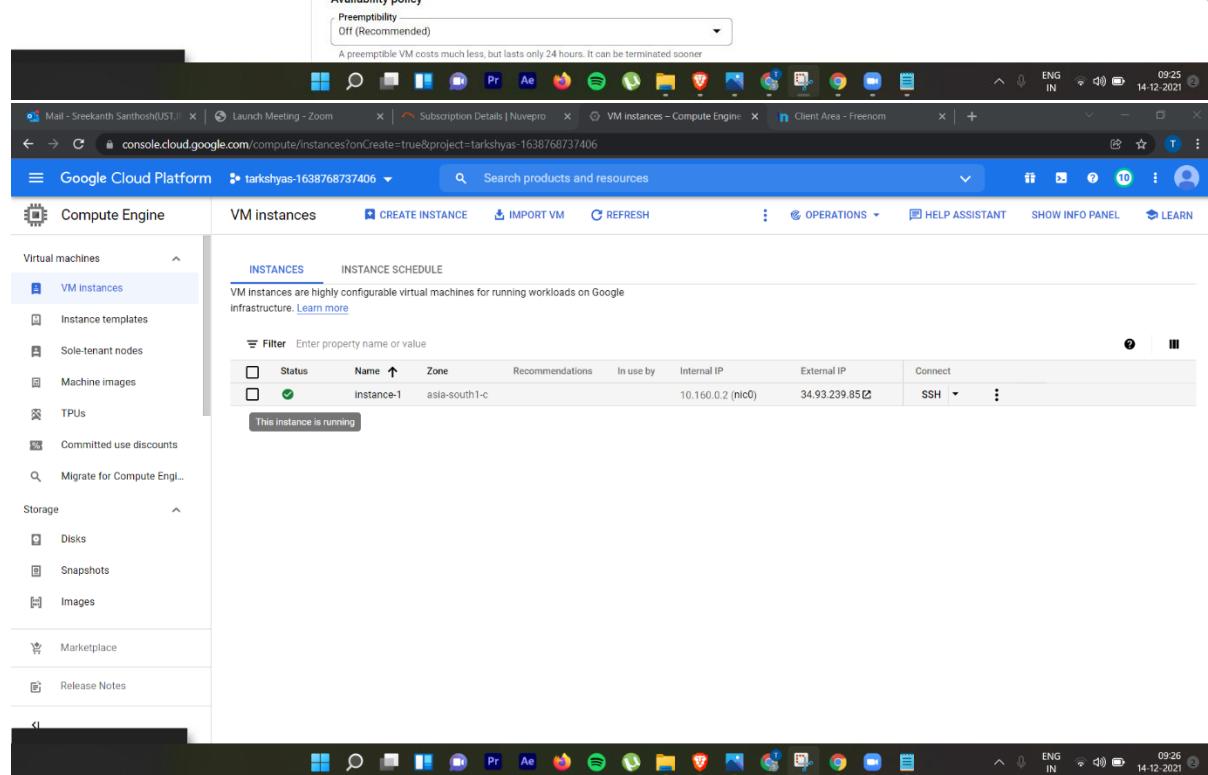
Reserve new static IP Address



Add startup Script



The screenshot shows the Google Cloud Platform interface for creating a new VM instance. On the left, there's a sidebar with options like 'New VM instance', 'New VM instance from template', 'New VM instance from machine image', and 'Marketplace'. The main area is titled 'Create an instance' and includes sections for 'Deletion protection', 'Reservations', 'Automation' (with a startup script field containing: `sudo apt-get update
sudo apt-get install -y apache2
echo <h1>Welcome</h1> > /var/www/html/index.html`), 'Metadata', and 'Availability policy'. A monthly estimate of '\$30.58' is shown.



The second screenshot shows the 'VM instances' page in the Compute Engine section of the Google Cloud Platform. It lists a single instance named 'Instance-1' with status 'Running'. The instance is located in 'asia-south1-c' zone. Its internal IP is 10.160.0.2 (nic0) and external IP is 34.93.239.85. The 'Connect' button is set to SSH. The sidebar on the left includes sections for Virtual machines (VM Instances, Instance templates, Sole-tenant nodes, Machine images, TPUs, Committed use discounts, Migrate for Compute Eng...), Storage (Disks, Snapshots, Images), and Marketplace.

STEP 3: Create Cloud DNS

The screenshot shows the Google Cloud Platform interface for the Cloud DNS API. The title bar includes tabs for Mail, Launch Meeting - Zoom, Subscription Details | Nuvepro, Cloud DNS API - Marketplace, Client Area - Freenom, and a search bar. Below the header, the main content area has a title 'Cloud DNS API' with a subtitle 'Google Enterprise API'. A sub-subtitle 'Highly Available Global DNS Network' is present. A 'TRY THIS API' button is visible. At the bottom of this section are tabs for 'OVERVIEW' (which is selected), 'PRICING', and 'DOCUMENTATION'.

The screenshot shows the Google Cloud Platform Network services page. The left sidebar lists services: Load balancing, Cloud DNS (which is selected and highlighted in blue), Cloud CDN, Cloud NAT, Traffic Director, Service Directory, Cloud Domains, and Private Service Connect. The main content area is titled 'Create a DNS zone' and contains the following fields:

- Zone type:** A radio button group where 'Public' is selected.
- Zone name:** An input field containing 'asia-zone'.
- DNS name:** An input field containing 'srk10.99'.
- DNSSEC:** A dropdown menu set to 'Off'.
- Description:** A text input field.
- Cloud Logging:** A radio button group where 'Off' is selected.

At the bottom of the form, a note states: 'After creating your zone, you can add resource record sets and modify the networks your zone is visible on.'

Sreekanth Santhosh-198642

The screenshot shows two consecutive screenshots of the Google Cloud Platform interface, both titled "cloud dns".

Screenshot 1: Zone details

The left sidebar shows "Network services" with "Cloud DNS" selected. The main area displays "Zone details" for "asia-zone". The configuration includes:

- DNS name: srk10.gq.
- Type: Public
- DNSSEC: Off
- Cloud Logging: Off

The "RECORD SETS" section shows the following table:

DNS name	Type	TTL (seconds)	Routing policy
srk10.gq.	SOA	21600	Default
srk10.gq.	NS	21600	Default

A success message box is visible: "DNS zone 'asia-zone' has been successfully created. Configure your domain at your registrar to make it accessible." with "LEARN MORE" and "X" buttons.

Screenshot 2: Resource record set details

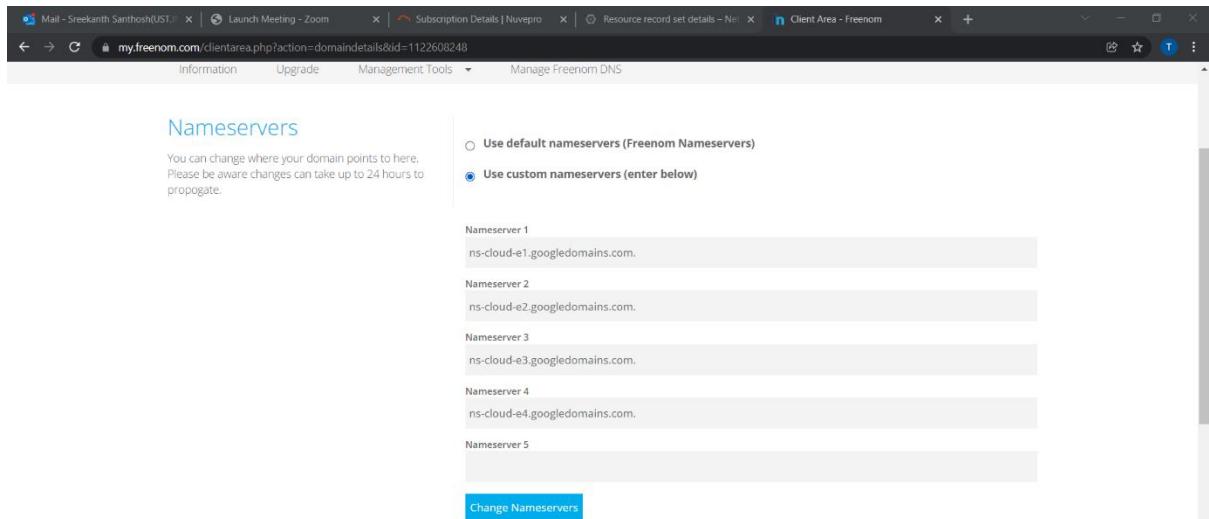
The left sidebar shows "Network services" with "Cloud DNS" selected. The main area displays "Resource record set details" for "srk10.gq./NS". The configuration includes:

- DNS name: srk10.gq.
- Type: NS
- TTL(seconds): 21600

The "Data" section shows the following table:

Data
ns-cloud-e1.googledomains.com.
ns-cloud-e2.googledomains.com.
ns-cloud-e3.googledomains.com.
ns-cloud-e4.googledomains.com.

A success message box is visible: "DNS zone 'asia-zone' has been successfully created. Configure your domain at your registrar to make it accessible." with "LEARN MORE" and "X" buttons.



Welcome



Create Resource Types A and CNAME

The screenshot shows the Google Cloud Platform Cloud DNS interface for creating a new record set. The left sidebar lists network services: Load balancing, Cloud DNS (selected), Cloud CDN, Cloud NAT, Traffic Director, Service Directory, Cloud Domains, and Private Service Connect. The main panel is titled 'Create record set' and shows the following fields:

- DNS Name:** .srk10.gq.
- Resource Record Type:** A (selected)
- TTL:** 5
- TTL Unit:** minutes
- Routing Policy:** Default record type (selected)
- IPv4 Address:** 34.93.239.85
- CREATE** button

The status bar at the bottom indicates the system language is English (ENG) and the date is 14-12-2021.

Sreekanth Santhosh-198642

The screenshot shows the Google Cloud Platform Cloud DNS interface. A modal window titled 'Create record set' is open. In the 'DNS Name' field, 'www.srk10.gq' is entered. The 'Resource Record Type' is set to 'CNAME'. The 'TTL' is set to '5' and 'TTL Unit' is 'minutes'. Under 'Routing Policy', 'Default record type' is selected. The 'Canonical name' field contains 'srk10.gq'. A success message at the bottom of the modal states: 'Resource record set 'srk10.gq'' has been successfully created in zone 'asia-zone''. The background shows the Cloud DNS service page with other options like Load balancing, Cloud CDN, Cloud NAT, Traffic Director, Service Directory, Cloud Domains, and Private Service Connect.

Welcome

9.Load balancing – http and tcp

A Cloud Load Balancer distributes user traffic across instances of an application in single region or multiple regions.

STEP 1: Create an Instance Template

The screenshot shows the 'Create an instance template' interface in the Google Cloud Platform. The left sidebar under 'Compute Engine' has 'VM Instances' selected. The main form is titled 'Create an instance template'. It includes fields for 'Name' (set to 'instance-template-1'), 'Labels' (empty), and 'Machine configuration' (set to 'GENERAL-PURPOSE' machine family, 'E2' series, 'e2-medium' type, 1 vCPU, 4 GB memory). A 'CREATE' button is at the bottom.

Select the Boot Disk as Ubuntu

The screenshot shows the 'Create an instance template' interface with the 'Boot disk' configuration dialog open. Under 'Container', 'Operating system' is set to 'Ubuntu'. Under 'Boot disk', 'Type' is 'Balanced persistent disk', 'Size' is '10 GB', and 'Image' is 'Ubuntu 18.04 LTS'. A 'SELECT' button is visible at the bottom right of the dialog.

Type in the Startup Script as shown in the notepad

The screenshot shows the 'Create an instance template' page in the Google Cloud Platform. On the left, there's a sidebar with 'Compute Engine' selected under 'Virtual machines'. In the main area, there's a 'Reservations' section, an 'Automation' section with a 'Startup script' field containing the provided code, and a 'Metadata' section. A 'Monthly estimate' box is visible on the right. A modal window titled 'New Text Document - Notepad' is open, showing the exact startup script code.

Instance template has successfully created

The screenshot shows the 'Instance templates' page in the Google Cloud Platform. The sidebar on the left is identical to the previous screenshot. The main area displays a table of existing instance templates. One row is present in the table, labeled 'instance-template-1'. The table includes columns for Actions, Creation time, Disk type, Image, Machine type, Name, Placement policy, and In use by.

STEP 2: Create instance group

To create an Instance group, select one of the options:

- New managed instance group (stateless)**
For stateless serving and batch workloads.
Supports:
 - autoscaling, autohealing, auto-updating
 - multi-zone deployment
 - load balancing
- New managed instance group (stateful)**
For stateful workloads such as databases.
Supports:
 - disk and metadata preservation
 - autohealing and updating
 - multi-zone-deployment
 - load balancing
- New unmanaged instance group**
A group of VMs that you manage yourself.
Supports:
 - load balancing

Name: vm
Description (Optional):
Location: Single zone
Region: us-central1 (Iowa) Zone: us-central1-a
Specify port name mapping (Optional): instance-template-1
Instance template: instance-template-1
Number of instances: 2
Autoscaling: Use autoscaling to allow automatic resizing of this instance group for periods of high and low load. Autoscaling groups of instances
Autoscaling mode: Don't autoscale

Select autoscaling mode as: Don't Autoscale

AUTOSCALING
Use autoscaling to allow automatic resizing of this instance group for periods of high and low load. Autoscaling groups of instances
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
Use metrics and schedules to determine when to autoscale the group. Autoscaling always needs at least one signal: either a metric or a schedule.
Autoscaling policy and target utilization
CPU utilization: 60% (default)
+ Add new metric
+ Add new scaling schedule

Predictive autoscaling
Use predictive autoscaling to predict future capacity needs based on historical CPU load.
Learn more
Off
No predictive method is used. The autoscaler scales the group to meet current demand based on real-time metrics.
Optimize for availability
Predictive autoscaling improves availability by monitoring daily and weekly CPU load patterns and scaling out ahead of anticipated demand.
⚠️ To enable predictive autoscaling, your group needs to have had an autoscaling configuration for at least 3 days.

Cool down period

Create a Health check

The screenshot shows the 'Create an instance group' page in the Google Cloud Platform Compute Engine interface. The 'Protocol' is set to 'HTTP' and 'Port' to '80'. The 'Proxy protocol' is 'NONE'. The 'Request path' is '/'. The 'Check interval' is '10 seconds' and 'Timeout' is '5 seconds'. The 'Healthy threshold' is '2 consecutive successes' and the 'Unhealthy threshold' is '3 consecutive failures'. A 'Save and continue' button is visible at the bottom.

The screenshot shows the 'Instance groups' list page in the Google Cloud Platform Compute Engine interface. It displays a single instance group named 'vm' with 2 instances, created on Dec 13, 2021, at 1:40:07 PM UTC+05:30. The group is managed and uses 'instance-template-1'. The 'Autoscaling' status is 'Off: Target CPU utilization 60%'. The instance group is located in the 'us-central1-a' zone. A 'CREATE INSTANCE GROUP' button is visible at the top right.

Status	Name	Instances	Template	Group type	Creation time	Recommendation	Autoscaling	Zone	In Use By
●	vm	2	instance-template-1	Managed	Dec 13, 2021, 1:40:07 PM UTC+05:30		Off: Target CPU utilization 60%	us-central1-a	

STEP 3: Create a Load balancer (Select it from Network Services in Navigation Menu)

The screenshot shows two stacked screenshots of the Google Cloud Platform interface.

Top Screenshot: The user is in the "Instance groups" section under the "Compute Engine" menu. A context menu is open over the "Network services" item in the sidebar, with "Load balancing" selected. The main pane displays a table of instance groups, including one named "vm" with 2 instances.

Status	Name	Instances	Template	Group type	Creation time	Recommendation	Autoscaling	Zone	In Use By
OK	vm	2	instance-template-1	Managed	Dec 13, 2021, 1:40:07 PM UTC+05:30		Off. Target CPU utilization 60%	us-central1-a	

Bottom Screenshot: The user is in the "LOAD BALANCERS" tab of the "Network services" section. The sidebar shows "Load balancing" is selected. The main pane displays the "Load balancing" page, which includes a "CREATE LOAD BALANCER" button and instructions for editing forwarding rules.

First, we are creating HTTP Load Balancer

The screenshot shows the Google Cloud Platform Network services - Create a load balancer interface. On the left sidebar, under Network services, the 'Load balancing' option is selected. The main area displays three configuration options:

- HTTP(S) Load Balancing**: Layer 7 load balancing for HTTP and HTTPS applications. It includes 'Configure' (HTTP LB, HTTPS LB (Includes HTTP/2 LB)) and 'Options' (Internet-facing or internal, Single or multi-region). A 'START CONFIGURATION' button is present.
- TCP Load Balancing**: Layer 4 load balancing or proxy for applications that rely on TCP/SSL protocol. It includes 'Configure' (TCP LB, SSL Proxy, TCP Proxy) and 'Options' (Internet-facing or internal, Single or multi-region). A 'START CONFIGURATION' button is present.
- UDP Load Balancing**: Layer 4 load balancing for applications that rely on UDP protocol. It includes 'Configure' (UDP LB) and 'Options' (Internet-facing or internal, Single-region). A 'START CONFIGURATION' button is present.

Below the configuration sections, there is a note: "Please answer a few questions to help us select the right load balancing type for your application".

Internet facing or internal only

Do you want to load balance traffic from the Internet to your VMs or serverless services, or only between VMs in your network?

From Internet to my VMs or serverless services
 Only between my VMs

Advanced traffic management

HTTP(S) Load Balancer with Advanced Traffic Management PREVIEW
 Classic HTTP(S) Load Balancer

CONTINUE

The status bar at the bottom indicates: 198622-Aarshya..., ENG IN, 13:37, 13-12-2021.

Create a Backend Service

The screenshot shows the Google Cloud Platform interface for creating a new HTTP(S) load balancer. The left sidebar under 'Network services' has 'Load balancing' selected. The main panel shows the 'Backend configuration' step, which is highlighted with a blue border. A note states: 'Only backend services created for HTTP(S) Load Balancer with Advanced Traffic Management will be visible. Backend services created for the Classic HTTP(S) Load Balancer cannot be used.' Below this are three options: 'Host and path rules', 'Frontend configuration', and 'Review and finalize (optional)'. At the bottom of the main panel are 'CREATE' and 'CANCEL' buttons. A modal window titled 'Backend services & backend buckets' is overlaid, containing a search bar with 'Filter Type to filter' and 'No matches for '''. It also has 'CREATE A BACKEND SERVICE' and 'CREATE A BACKEND BUCKET' buttons. The system status bar at the bottom right shows the user's name '198622-Aarshya...' and the date '13-12-2021'.

The screenshot shows two consecutive steps in the Google Cloud Platform Network Services Load Balancing interface:

Step 1: Create backend service

- Name:** web-backend
- Description:** Lowercase, no spaces.
- Backend configuration:**
 - Host and path rules:** Host
 - Frontend configuration:** Frontend
- Protocol:** HTTP
- Named port:** http
- Timeout:** 30 seconds

Step 2: Backends

New backend

- Instance group:** vm
- Port numbers:** 5000

Create backend service

Backends

Regions: us-central1

New backend

- Instance group:** vm
- Port numbers:** 5000
- Balancing mode:** Utilization
- Maximum backend utilization:** 80 %
- Maximum RPS:** RPS
- Scope:** per instance
- Capacity:** 100 %

Create a Frontend Service

The screenshot shows the Google Cloud Platform interface for creating a new HTTP(S) load balancer. On the left sidebar, under 'Network services', 'Load balancing' is selected. In the main area, a 'New HTTP(S) load balancer' dialog is open. The 'Frontend configuration' tab is active. A modal window titled 'New Frontend IP and port' is displayed, containing the following fields:

- Name:** web-frontend
- Protocol:** HTTP
- IP version:** IPv4
- IP address:** Ephemeral
- Port:** 80

At the bottom of the modal, there are 'CANCEL' and 'DONE' buttons, and a large 'ADD FRONTEND IP AND PORT' button.

The screenshot shows the Google Cloud Platform interface for managing load balancers. The left sidebar shows 'Network services' with 'Load balancing' selected. The main area displays a table of load balancers:

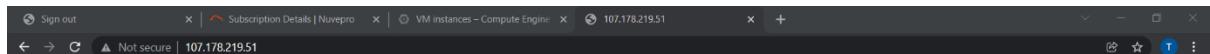
	Name	Load balancer type	Protocols	Region	Backends
<input type="checkbox"/>	web-loadbalancer	HTTP(S)	HTTP		1 backend service (1 instance group, 0 network endpoint groups)

A message at the bottom of the table reads: "To edit load balancing resources like forwarding rules and target proxies, go to the advanced menu."

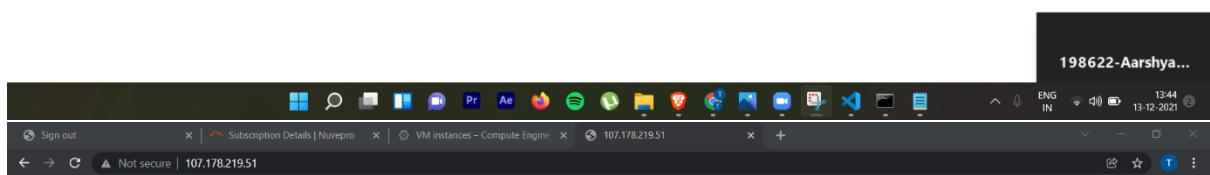
Go to VM Instances and check the External IP addresses

The screenshot shows the Google Cloud Platform Compute Engine VM Instances page. The left sidebar lists various Compute Engine management options like Snapshots, Images, Instance groups, VM Manager, Bare Metal Solution, and Settings. The main content area displays the 'INSTANCES' section with a table of running VMs. The table includes columns for Status, Name, Zone, Recommendations, In use by, Internal IP, External IP, and Connect. Two VM instances are listed:

Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
Green checkmark	vm-2bxh	us-central1-a		vm	10.128.0.4 (nic0)	107.178.219.51	SSH
Green checkmark	vm-qk88	us-central1-a		vm	10.128.0.3 (nic0)	35.223.95.39	SSH



Traffic from vm-2bxh



Traffic from vm-2bxh



STEP 4: Create a new TCP load balancer

The image consists of two side-by-side screenshots from the Google Cloud Platform (GCP) console.

Screenshot 1: Create an instance group

This screenshot shows the configuration of a new instance group named "tcp-healthcheck". The protocol is set to TCP with port 80. The health criteria are defined with a check interval of 10 seconds and a timeout of 5 seconds. The healthy threshold is 2 consecutive successes, and the unhealthy threshold is 3 consecutive failures.

Screenshot 2: New TCP load balancer

This screenshot shows the creation of a new TCP load balancer named "web-loadbalancer-tcp". It is configured to use the "us-central1 (Iowa)" region. The "Backend configuration" tab is selected, showing the "Backend service" name "web-loadbalancer-tcp" and the "Protocol" set to TCP. Under "Backends", a new backend is being added, pointing to the instance group "instance-group-1". A health check named "tcp-healthcheck1" is assigned to this backend.

Sreekanth Santhosh-198642

The screenshot displays two windows from the Google Cloud Platform Network services - Load balancing interface.

Top Window: New TCP load balancer

This window shows the configuration steps for creating a new TCP load balancer:

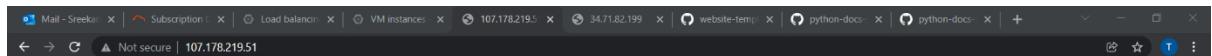
- Frontend configuration:** Name: web-loadbalancer-tcp, Region: us-central1 (Iowa).
- Backend configuration:** Backend configuration is selected.
- Frontend configuration:** Frontend configuration is selected.
- Review and finalize (optional):** This step is optional.

Bottom Window: Load balancing - Network services

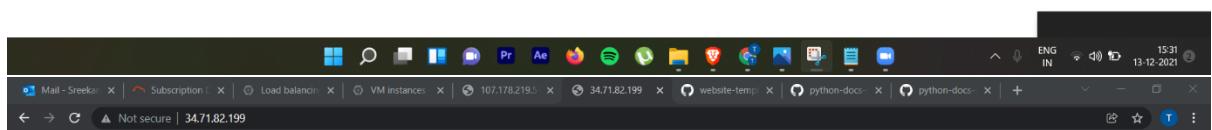
This window lists existing load balancers:

Name	Type	Protocols	Region	Backends
web-loadbalancer-tcp	Network (backend service-based)	TCP	us-central1	1 regional backend service (1 instance group, 0 network endpoint groups)

A message at the bottom states: "All backend services are healthy".



Traffic from instance-group-1-ptrp



Traffic from instance-group-1-cs79



10.GKE

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

Kubernetes Engine

Kubernetes clusters

Clusters

Workloads

Services & Ingress

Applications

Configuration

Storage

Object Browser

Migrate to containers

Config Management

Marketplace

Release Notes

CREATE

REFRESH

Kubernetes Engine

Kubernetes clusters

Containers package an application so it can easily be deployed to run in its own isolated environment. Containers are run on Kubernetes clusters. [Learn more](#)

CREATE DEPLOY CONTAINER TAKE THE QUICKSTART

Mail - Sreekanth Santhosh... | Launch Meeting - Zoom... | Subscription Details | Nuv... | VM instances - Compute Engine | VPN - Hybrid Connectivity | Kubernetes clusters - Kube... | +

Google Cloud Platform tarkshyas-1638768737406 Search products and resources

10:22 14-12-2021

Create cluster

Select the cluster mode that you want to use.

COMPARE

GKE Standard

A pay-per-node Kubernetes cluster where you configure and manage your nodes. [Learn more](#)

CONFIGURE

GKE Autopilot

A pay-per-Pod Kubernetes cluster where GKE manages your nodes with minimal configuration required. [Learn more](#)

CANCEL

Mail - Sreekanth Santhosh... | Launch Meeting - Zoom... | Subscription Details | Nuv... | VM instances - Compute Engine | VPN - Hybrid Connectivity | Kubernetes clusters - Kube... | +

Google Cloud Platform tarkshyas-1638768737406 Search products and resources

10:23 14-12-2021

The screenshot shows two windows side-by-side. The top window is titled 'Create a Kubernetes cluster' and is in the 'Cluster basics' step. It shows a sidebar with 'NODE POOLS' (default-pool selected), 'CLUSTER' (Automation, Networking, Security, Metadata, Features), and a 'Name' field set to 'cluster-1'. The bottom window shows the 'Kubernetes Engine' dashboard with 'Clusters' selected, displaying a table with one row: 'cluster-1' (Status: green checkmark, Location: us-central1-c, Number of nodes: 3, Total vCPUs: 6, Total memory: 12 GB). Below this is a 'Terminal' window titled '(tarkshyas-1638768737406)' showing a Docker run command for nginx:latest.

Status	Name	Location	Number of nodes	Total vCPUs	Total memory	Notifications	Labels
<input checked="" type="checkbox"/>	cluster-1	us-central1-c	3	6	12 GB	-	-

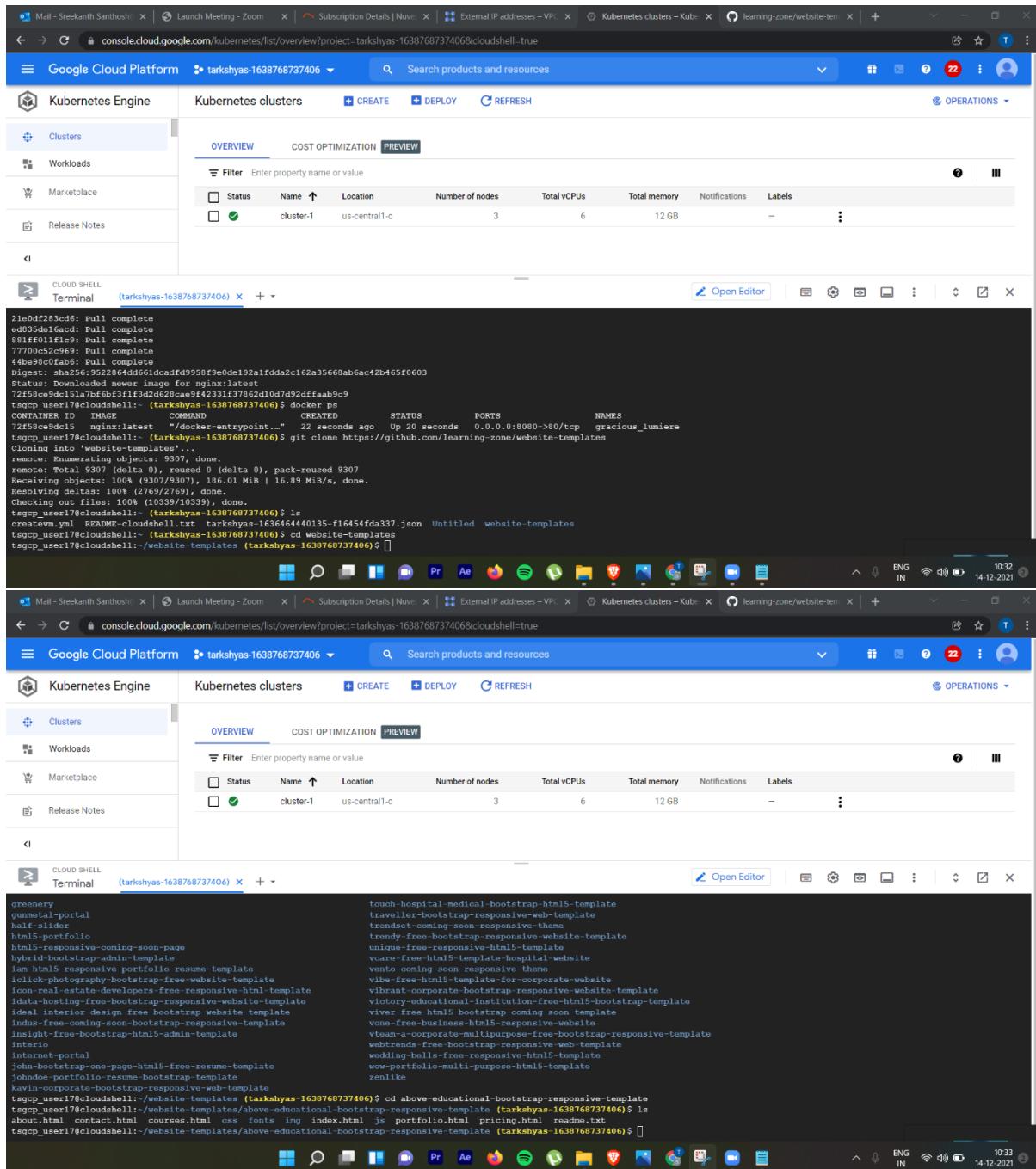
```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to tarkshyas-1638768737406.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
tsgcp_user17@cloudshell:~ (tarkshyas-1638768737406)$ docker run -d -p 8080:80 nginx:latest
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
e5ae68f74026: Pull complete
21e0df283cd6: Pull complete
ed835de16acd: Pull complete
88935a2a2a2c: Pull complete
77700c52e368: Pull complete
44be98e0fabe: Pull complete
Digest: sha256:9522864dd61dcadfd9958f9e0de192a1ffdda2c162a35660ab6ac42b465f0603
Status: downloaded newer image for nginx:latest
72f58c9d1c151a7bfef3ff3f1f3dd2d628cae9f42331f37862d10d7d92dffaa9c9
tsgcp_user17@cloudshell:~ (tarkshyas-1638768737406)$
```

Sreekanth Santhosh-198642

Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to **tarkshyas-1638768737406**.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
tsgcp_user17@cloudshell:~ (tarkshyas-1638768737406)\$ 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
881f0111fc19: Pull complete
77700c52e969: Pull complete
44be98c0fab6: Pull complete
Digest: sha256:9522864dd61dcadfd9958f9e0de192a1ffda2c162a3566ab6ac42b465f0603
Status: Downloaded newer image for nginx:latest
72f58c9dc15a70f6bf3f1f3dd628cae9f42331f37862d10d7d92dfafab9c9
tsgcp_user17@cloudshell:~ (tarkshyas-1638768737406)\$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
72f58c9dc15 nginx:latest "/docker-entrypoint..." 22 seconds ago Up 20 seconds 0.0.0.0:8080->80/tcp gracious_lumiere
tsgcp_user17@cloudshell:~ (tarkshyas-1638768737406)\$

Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to **tarkshyas-1638768737406**.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
tsgcp_user17@cloudshell:~ (tarkshyas-1638768737406)\$ 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
881f0111fc19: Pull complete
77700c52e969: Pull complete
44be98c0fab6: Pull complete
Digest: sha256:9522864dd61dcadfd9958f9e0de192a1ffda2c162a3566ab6ac42b465f0603
Status: Downloaded newer image for nginx:latest
72f58c9dc15a70f6bf3f1f3dd628cae9f42331f37862d10d7d92dfafab9c9
tsgcp_user17@cloudshell:~ (tarkshyas-1638768737406)\$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
72f58c9dc15 nginx:latest "/docker-entrypoint..." 22 seconds ago Up 20 seconds 0.0.0.0:8080->80/tcp gracious_lumiere
tsgcp_user17@cloudshell:~ (tarkshyas-1638768737406)\$ git clone https://github.com/learning-zone/website-templates
Cloning into 'website-templates'...
remote: Enumerating objects: 9307, done.
Receiving objects: 94% (7818/9307), 160.43 MiB | 17.32 MiB/s
10:31 14-12-2021 ENG IN

Sreekanth Santhosh-198642



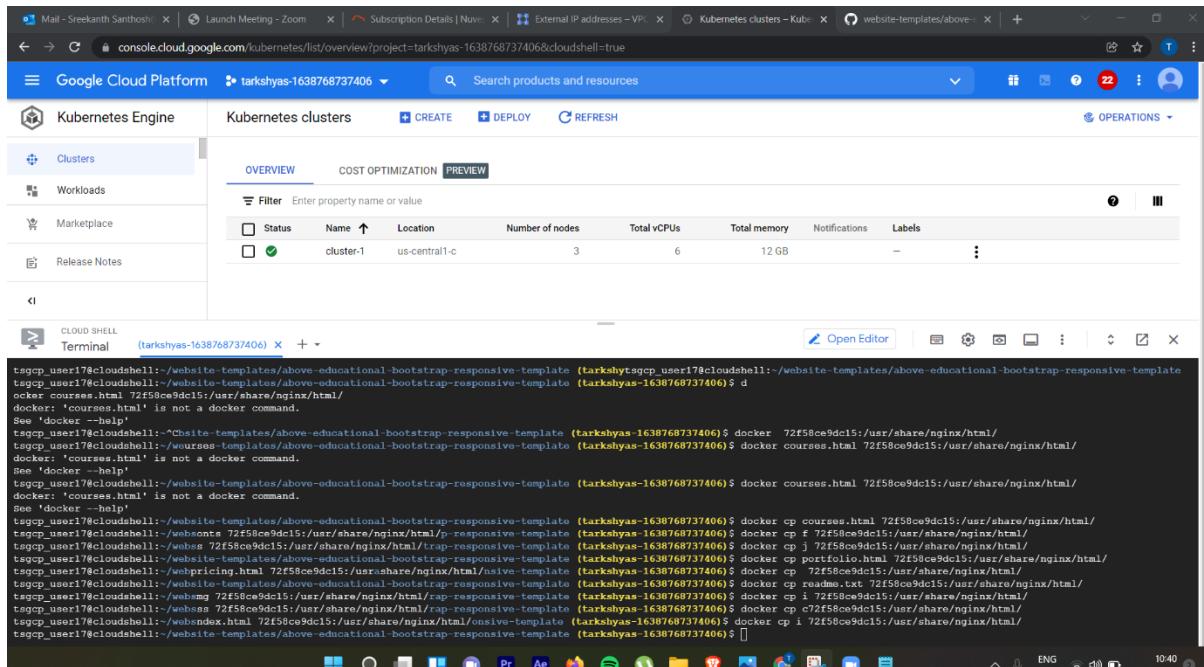
The screenshot shows two instances of the Google Cloud Platform interface. Both instances are for the project 'tarkshyas-1638768737406'. The top instance displays the 'Kubernetes Engine' section, specifically the 'Clusters' tab. It lists a single cluster named 'cluster-1' located in 'us-central1-c' with 3 nodes, 6 vCPUs, and 12 GB of memory. The bottom instance shows the 'Cloud Shell' terminal for the same project. In the terminal, several commands are run:

```
21e0df283cd6: Full complete
ed835de1accd: Full complete
881f011fc1c9: Full complete
77700c52e969: Full complete
44be98c0f016: Full complete
bb0d9a3a85264f4a... dcdaf9d958f9e0d... 192a1fdd... 2c162a35668ab6ac42b465f0603
Status: Downloaded newer image for nginx:latest
72f58c9dc15170fcbf3f1fd2d628cae9f42331f7862d10d7d92ffab9c9
tsgcp_user17@cloudshell: ~(tarkshyas-1638768737406)$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
72f58c9dc15 nginx:latest "/docker-entrypoint..." 22 seconds ago Up 20 seconds 0.0.0.0:8080->80/tcp gracius_lumiere
tsgcp_user17@cloudshell: ~(tarkshyas-1638768737406)$ git clone https://github.com/learning-zone/website-templates
Cloning into 'website-templates'...
remote: Enumerating objects: 930, done.
remote: Counting objects: 100% (930/930), pack-rotated 930%
Resolving deltas: 100% (2769/2769), done.
Checking out files: 100% (1039/1039), done.
tsgcp_user17@cloudshell: ~(tarkshyas-1638768737406)$ ls
readme.yml README-cloudshell.txt tarkshyas-163646440135-f16454fd337.json Untitled website-templates
tsgcp_user17@cloudshell: ~(tarkshyas-1638768737406)$ cd website-templates
tsgcp_user17@cloudshell:~/website-templates (tarkshyas-1638768737406)$ 
```

Following this, another terminal session is shown with the command:

```
tsgcp_user17@cloudshell: ~/(tarkshyas-1638768737406)$ ls
greenery
portfolio-portal
half-slider
html5-portfolio
html5-responsive-coming-soon-page
hybrid-bootstrap-admin-template
iam-html5-responsive-portfolio-free-website-template
iclick-photography-bootstrap-free-website-template
icon-real-estate-developers-free-responsive-html-template
idm-meeting-free-bootstrap-responsive-website-template
ideal-interior-design-free-bootstrap-website-template
indus-free-coming-soon-bootstrap-responsive-template
insight-free-bootstrap-html5-admin-template
interario
internet-portal
john-bootstrap-one-page-html5-free-resume-template
john doe portfolio resume bootstrap template
kavin-corporate-bootstrap-responsive-web-template
tsgcp_user17@cloudshell:~/website-templates (tarkshyas-1638768737406)$ cd above-educational-bootstrap-responsive-template
tsgcp_user17@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1638768737406)$ ls
about.html contact.html courses.html css fonts img index.html js portfolio.html pricing.html readme.txt
tsgcp_user17@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template (tarkshyas-1638768737406)$ 
```

Sreekanth Santhosh-198642

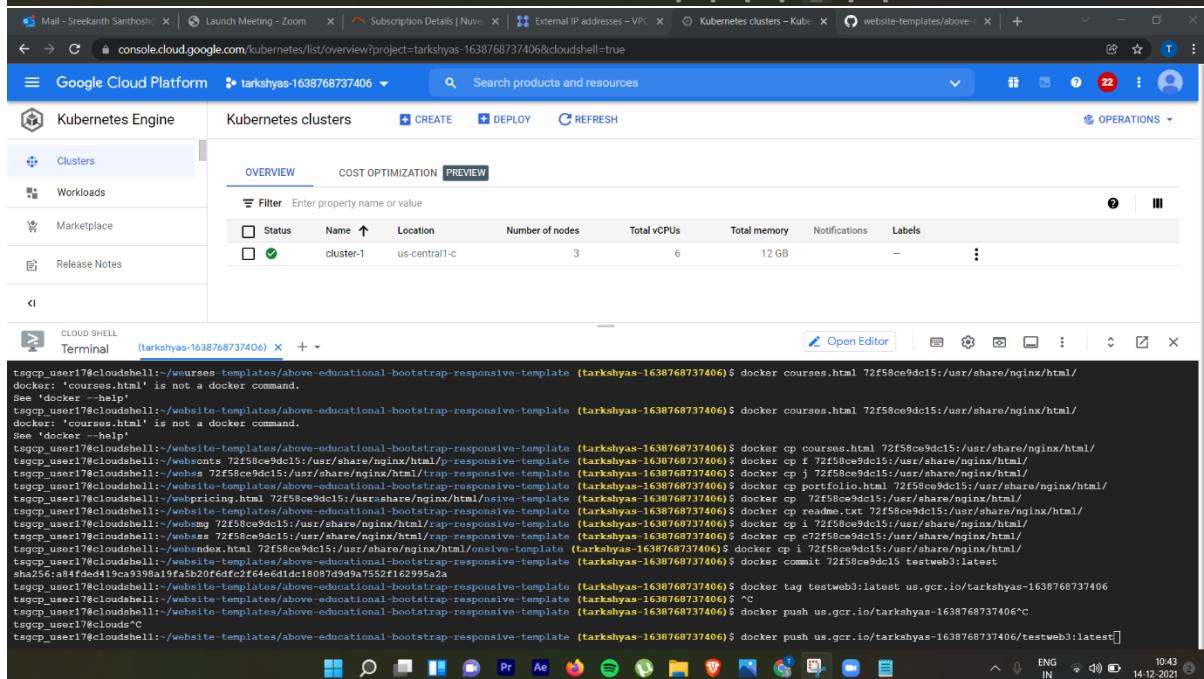


The screenshot shows the Google Cloud Platform (GCP) interface for the Kubernetes Engine. On the left, there's a sidebar with 'Clusters' selected. The main area displays a table for 'Kubernetes clusters'. One cluster, 'cluster-1', is listed with the following details:

Status	Name	Location	Number of nodes	Total vCPUs	Total memory	Notifications	Labels
Green checkmark	cluster-1	us-central1-c	3	6	12 GB	-	-

Below the table is a terminal window titled '(tarkshyas-1638768737406)'. It contains several commands related to Docker and Nginx, such as:

```
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyasgcp_user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template]
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker courses.html 72f58ce9dc15:/usr/share/nginx/html/
docker: 'courses.html' is not a docker command.
See 'docker --help'.
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker courses.html 72f58ce9dc15:/usr/share/nginx/html/
docker: 'courses.html' is not a docker command.
See 'docker --help'.
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp courses.html 72f58ce9dc15:/usr/share/nginx/html/
docker: 'courses.html' is not a docker command.
See 'docker --help'.
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp courses.html 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp f 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp portfolio.html 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp g 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp homepage.txt 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp readme.txt 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp c72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp i 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp j 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp k 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp l 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp m 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp n 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp o 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp p 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp q 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp r 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp s 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp t 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp u 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp v 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp w 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp x 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp y 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker cp z 72f58ce9dc15:/usr/share/nginx/html/
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker commit 72f58ce9dc15 testweb3:latest
sha256:a4f4ded419ca399a19fa5b20f6dfc2f64ed1dc18087d9a97552f162995a2a
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker tag testweb3:latest us.gcr.io/tarkshyas-1638768737406
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ ^C
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker push us.gcr.io/tarkshyas-1638768737406
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker push us.gcr.io/tarkshyas-1638768737406/testweb3:latest
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker push us.gcr.io/tarkshyas-1638768737406/testweb3:latest]
```



The second screenshot shows the same GCP interface and terminal window. The terminal output continues from the previous session, showing the creation of a Docker container and pushing it to a Google Container Registry (GCR) repository. The final command shown is:

```
tsgcg user1@cloudshell:~/website-templates/above-educational-bootstrap-responsive-template [tarkshyas-1638768737406]$ docker push us.gcr.io/tarkshyas-1638768737406/testweb3:latest
```

Sreekanth Santhosh-198642

Expose a deployment

Port mapping

Port *	Target port	Protocol
80	8081	TCP

+ ADD PORT MAPPING

Service type: Load balancer

Service name: nginx-1-service

EXPOSE VIEW YAML

* Indicates required field

About Kubernetes services

Services allow your deployments to receive traffic. Services can be exposed in different ways.

- Load balancer** - create a fixed external IP to route traffic to your deployment
- Node port** - create a fixed external IP to route traffic to your deployment
- Cluster IP** - create a custom virtual IP range to route traffic to your deployment.

Service details

Cluster: cluster-1
Namespace: default
Labels: app:nginx-1
Logs: nginx-1
Type: LoadBalancer
External endpoints: 34.72.145.125:80

Load Balancer

Cluster IP: 10.8.14.70
Load balancer IP: 34.72.145.125
Load balancer: a9ee8fee843f4495da51985a0959c6da

Deployments

Name	Status	Pods
nginx-1	OK	3/3

Serving pods

Name	Status	Endpoints	Restarts	Created on
nginx-1-754ddbcd6c-8bqsh	Running	10.4.0.5	0	Dec 14, 2021, 10:49:23 AM

Load balancer

This service has a fixed external IP to route traffic to your application. The IP address is externally facing. Visit the address to see the deployment.

Suggested next steps

- Scale the deployment by changing the number of replicas in the deployment.
- Perform a rolling update to change the deployment image.
- Deploy a stateful application to your cluster.

