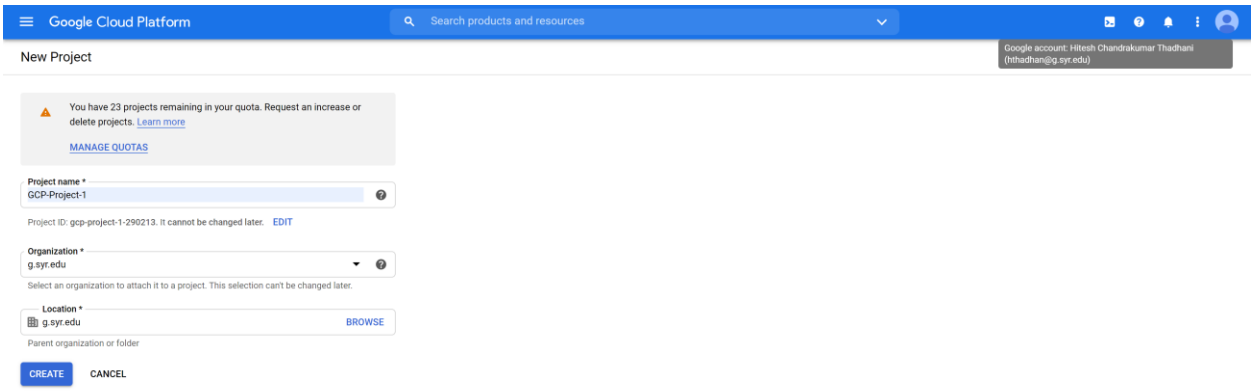
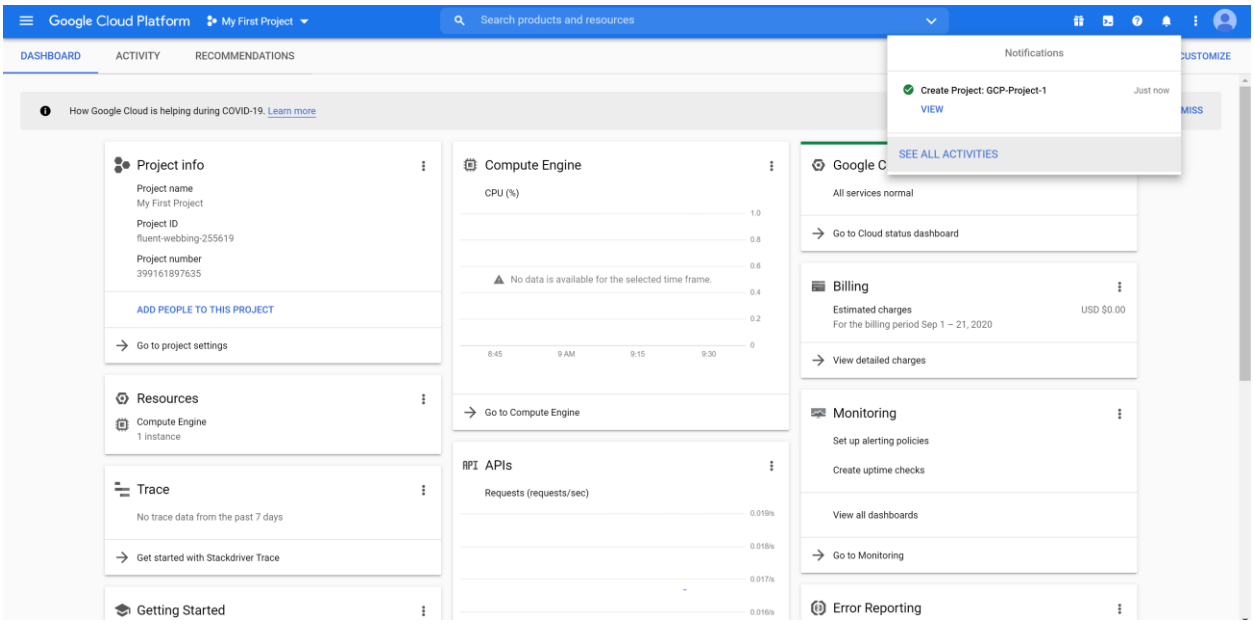
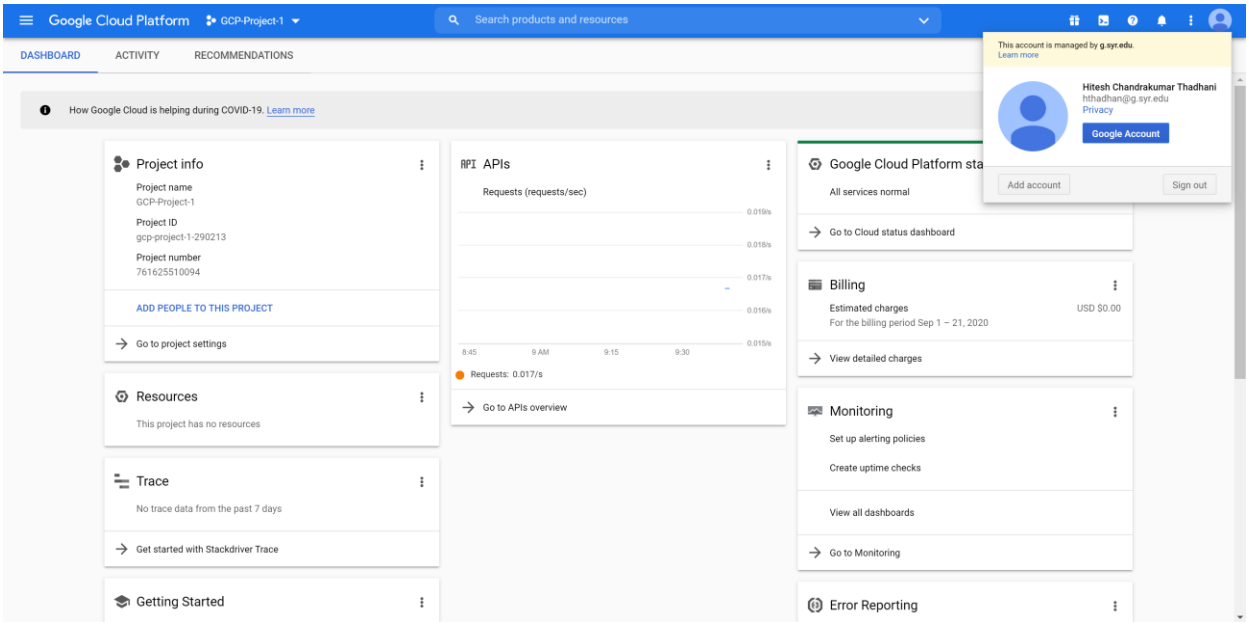
3. Cloud Shell

Created a New Cloud Project GCP-Project-1 for the lab

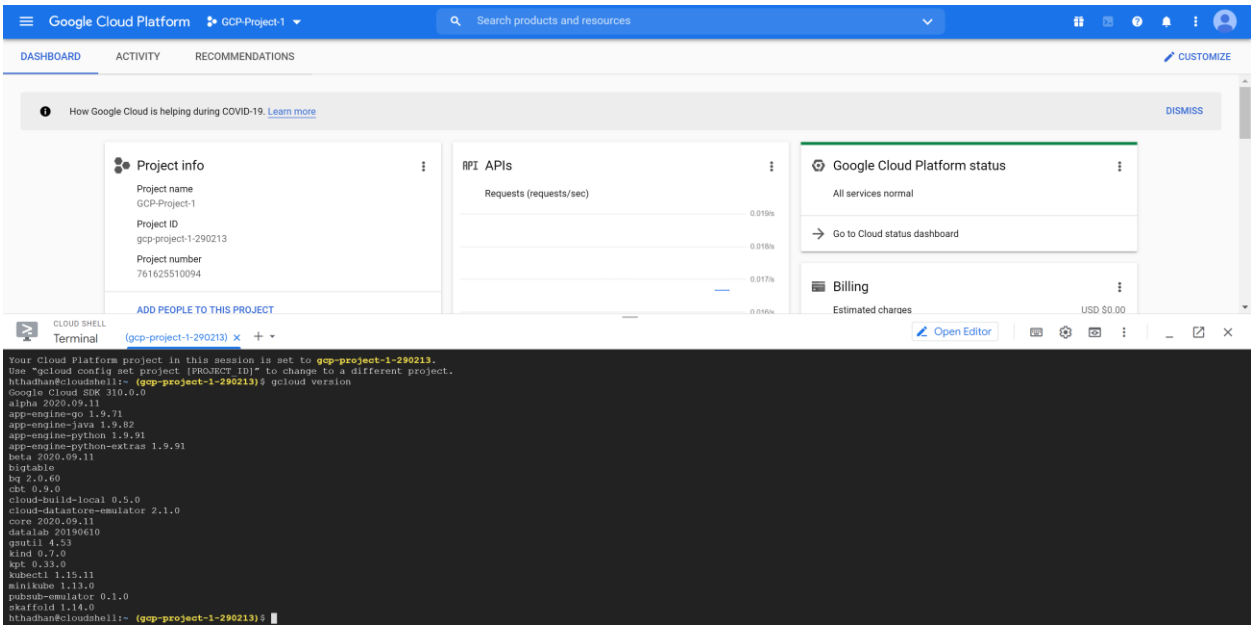




GCP-Project-1 Cloud Project Created



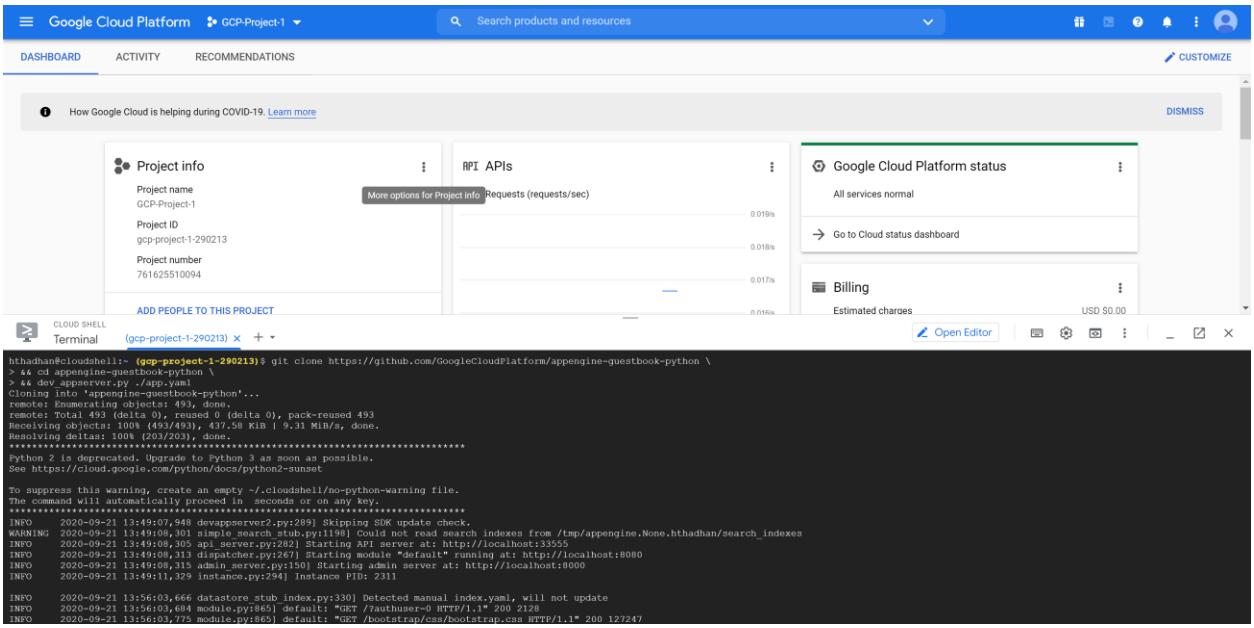
Started the gcloud shell using the Activate Cloud Shell option on the top-right



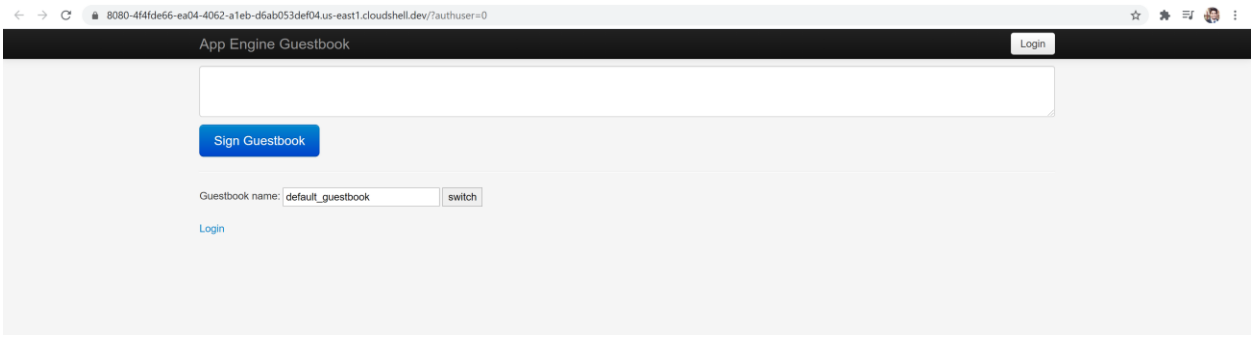
Preview and Deploy an App Engine application

Cloned the github repository containing a sample app and locally running it using the App Deployment server. This local running of the app is being viewed in the web browser using the

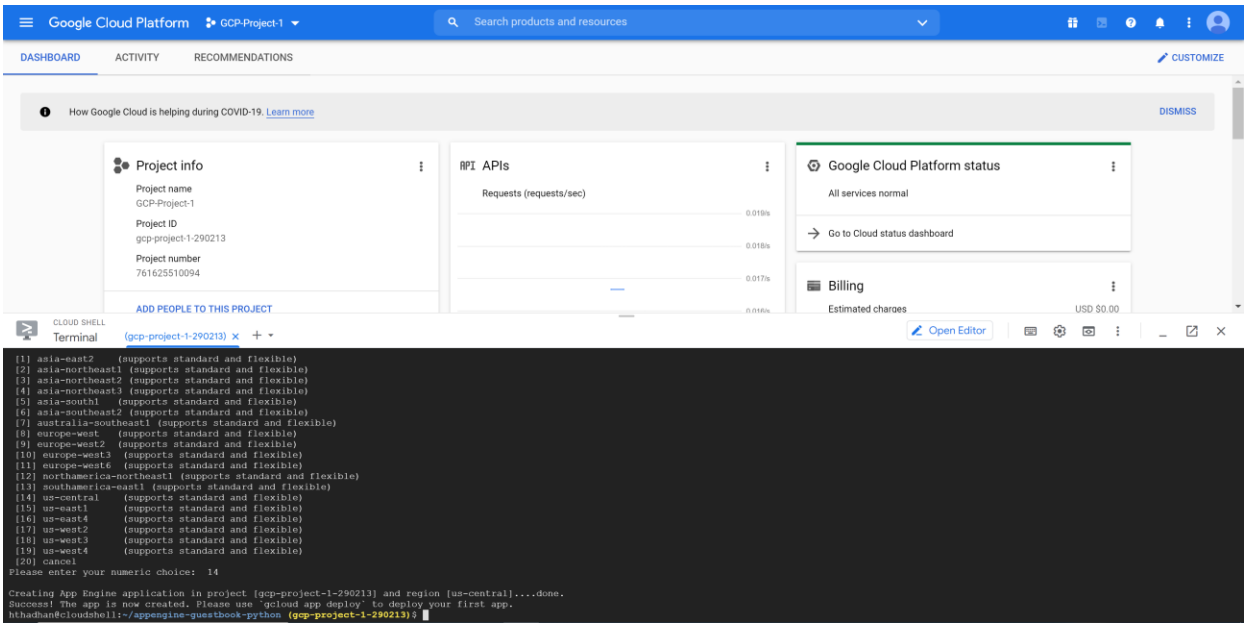
Web Preview button on the port 8080 so that before deploying this App online on the internet we can view and make any final changes.



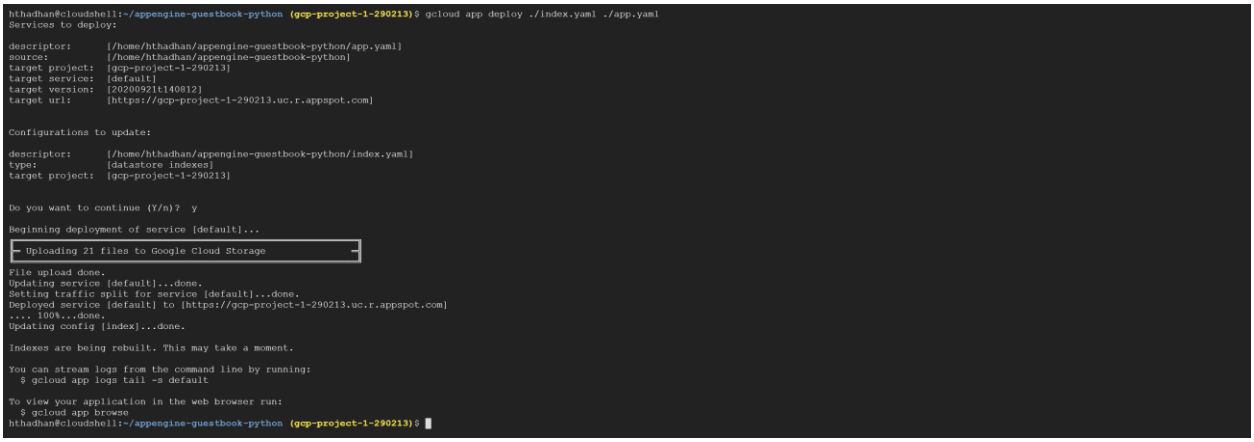
Web Preview on the port 8080 for the app.



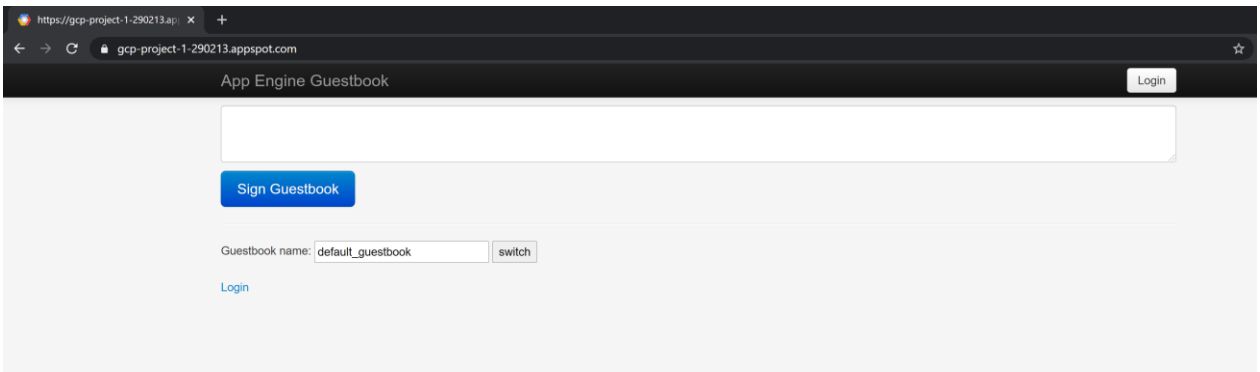
Attached the App Engine to the project created earlier for the lab.



- Deployed the App using App Deployment server to be accessible from the internet. After the app is deployed, the web browser link to access the app is shown in the target url along with the project name.

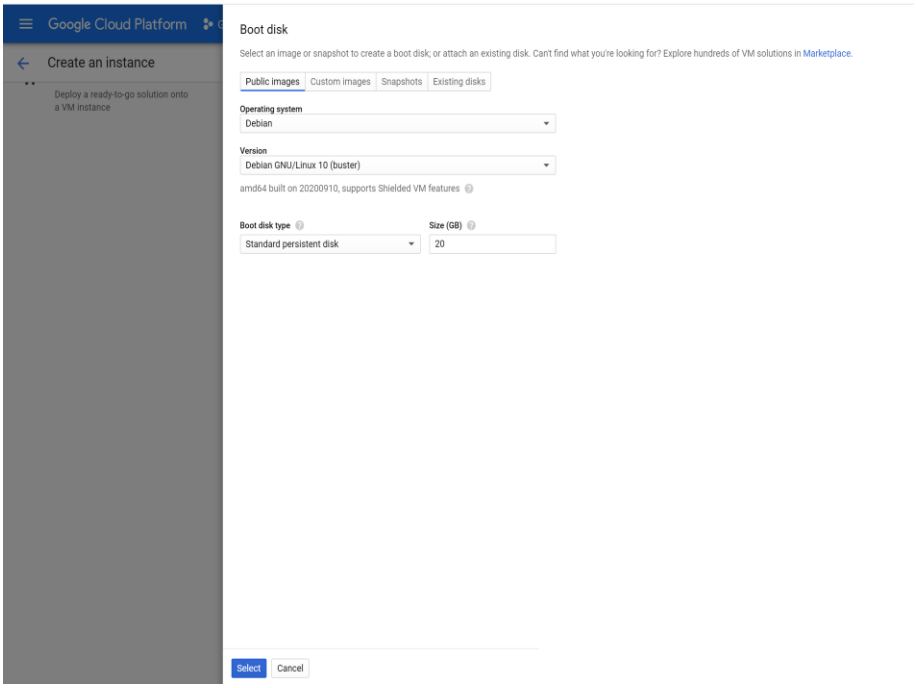


- Previewing the application deployed on the GCP for this project using the Web Browser using the target url with the ProjectID included in it.

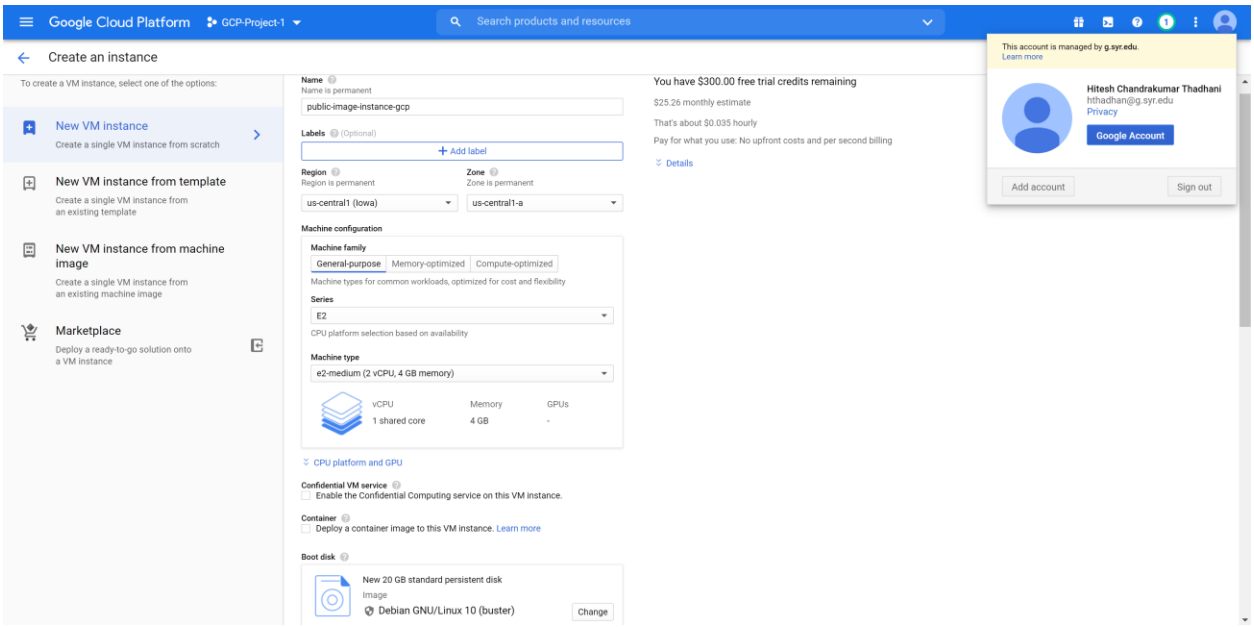


Section 4: Creating VM instances

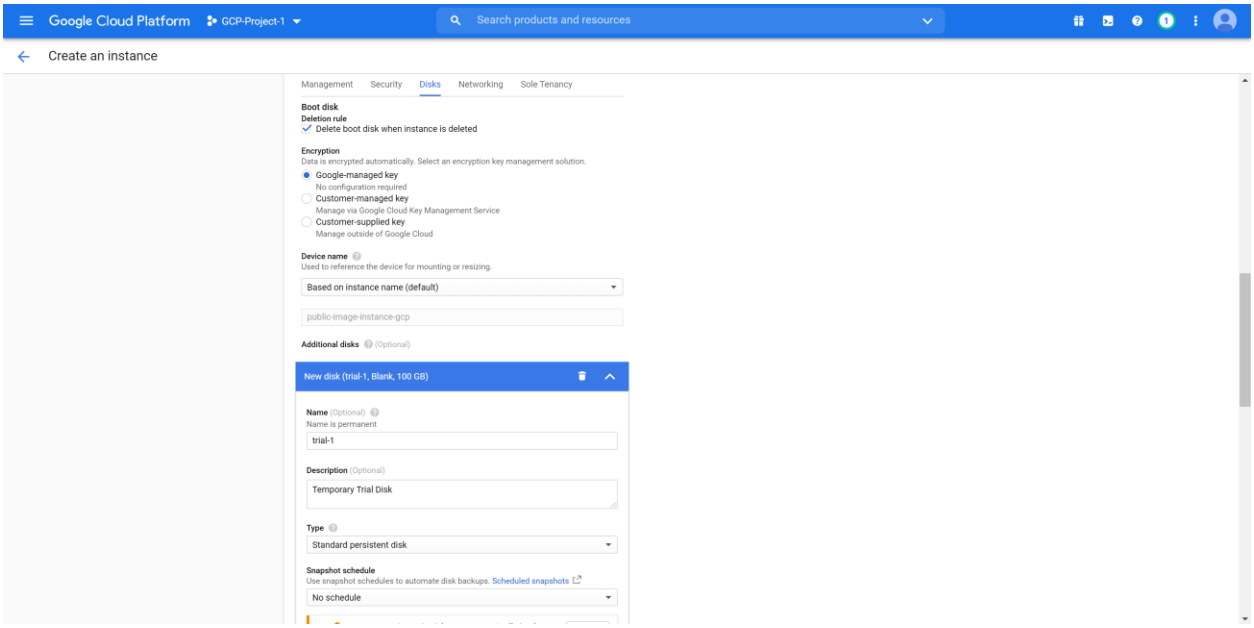
A. Creating an instance from a public image

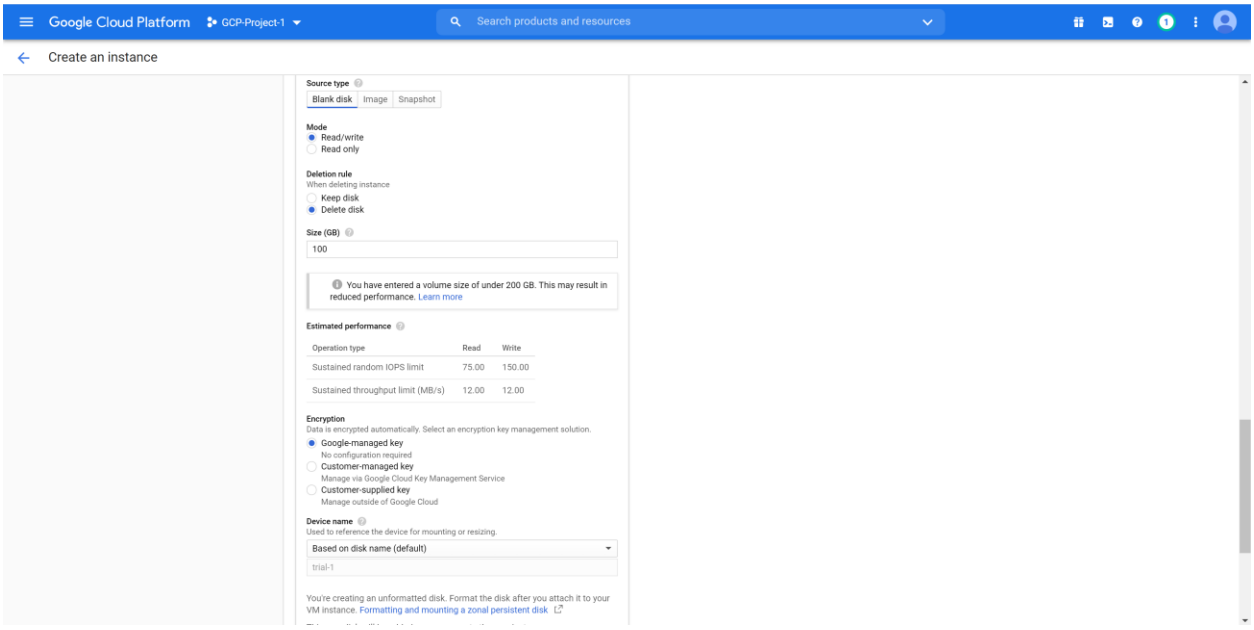


There are multiple public OS images available including Linux flavors like Debian along with Windows flavors SQL Server for Windows Operating systems. I selected Public Image of Debian Linux and boot disk space of 20GB is selected from the available OS images. These public images are available for everyone to use with all the preconfigured settings and it consists of all the required files to boot the OS.

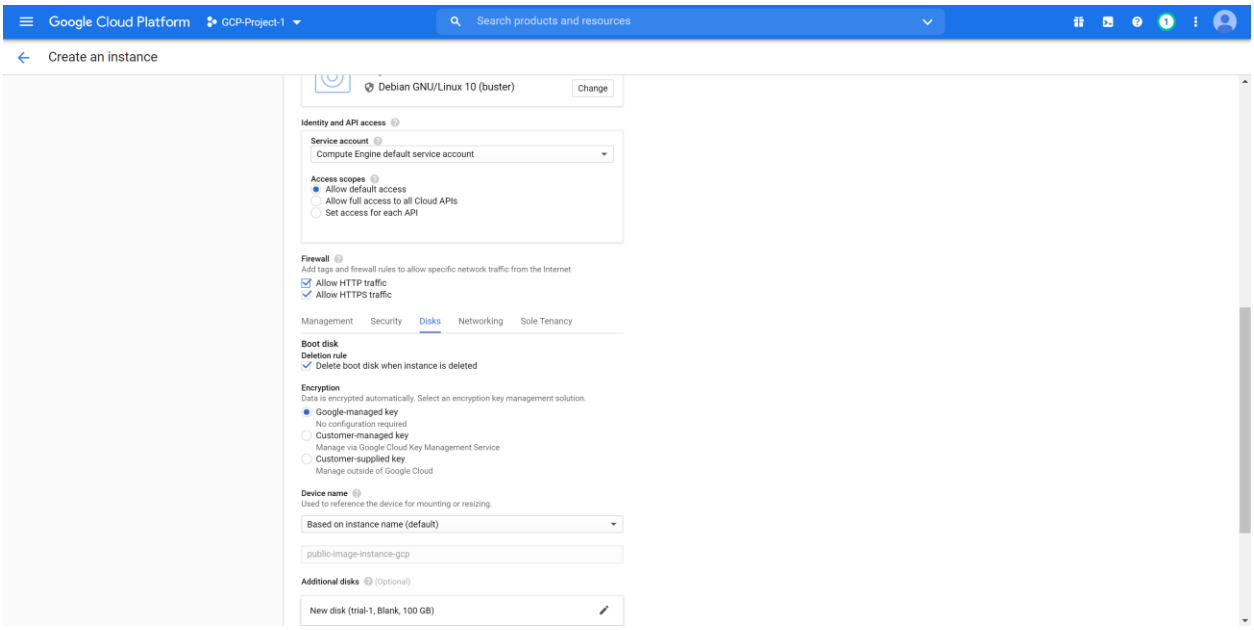


Additional Disk is selected for Temporary Trial purpose of 100GB with Read/Write configuration for allowing data to be written on this hard disk to be used as an additional hard disk for the VM and using Google Managed Key for the encryption to be managed by Google. The data before being written to the disk is encrypted by Goggle when Goggle Managed key option is selected and is automatically decrypted when reading the data. All this encryption and decryption of the data is taken care of by the Goggle to maintain the security. The temporary disk of 100GB would be deleted by default when the VM instance is deleted using the Delete Role Delete Disk so that the disk is deleted to free up space automatically when the VM is deleted.

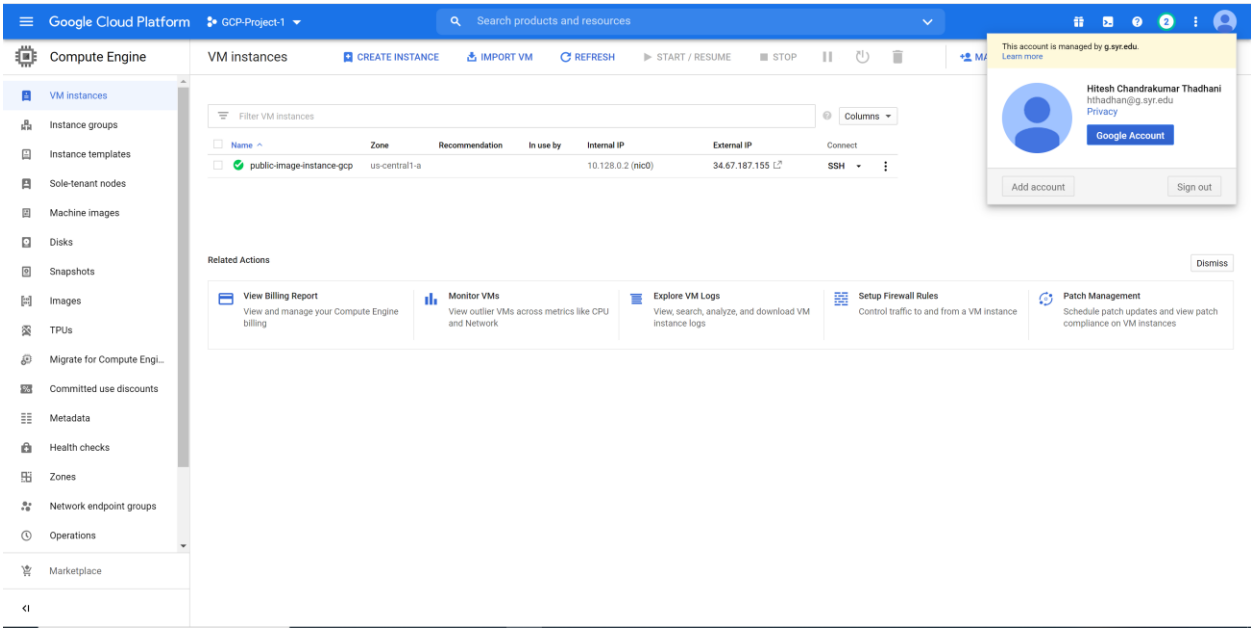




Allow HTTP and HTTPS traffic is selected so that the VM can communicate over the default port 80 for HTTP and 443 for HTTPS. To access the instance over the internet, HTTP and HTTPS protocols would be used. All the traffic coming to the instance would be coming to this port 80 and port 443. This creates a rule to allow the HTTP and HTTPS traffic coming to the VM over the internet.



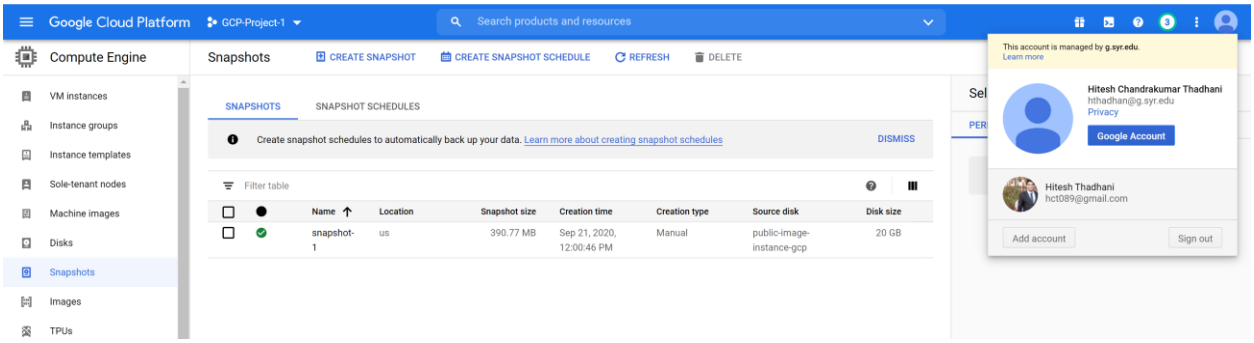
The instance created using the Public image of the OS is up and running which is seen below.



B. Creating an instance from a snapshot

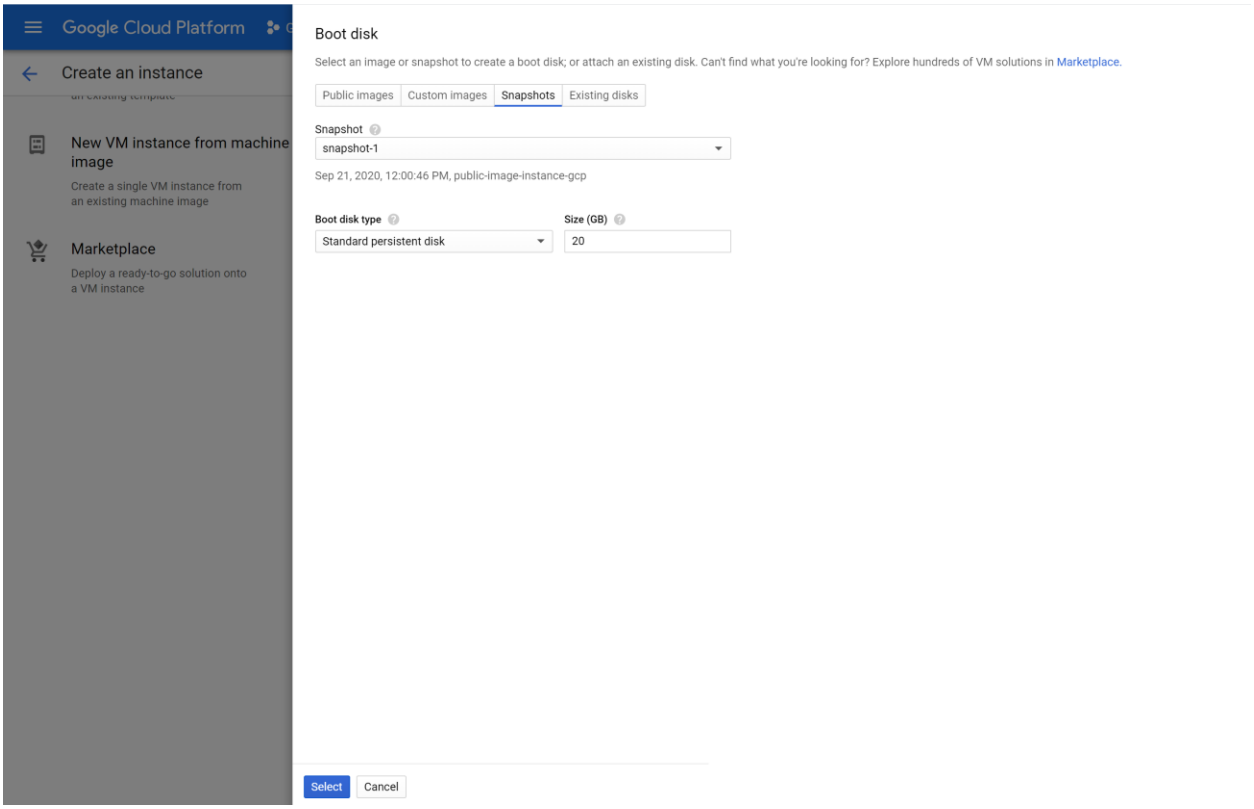
i. Create a snapshot before creating an instance from it –

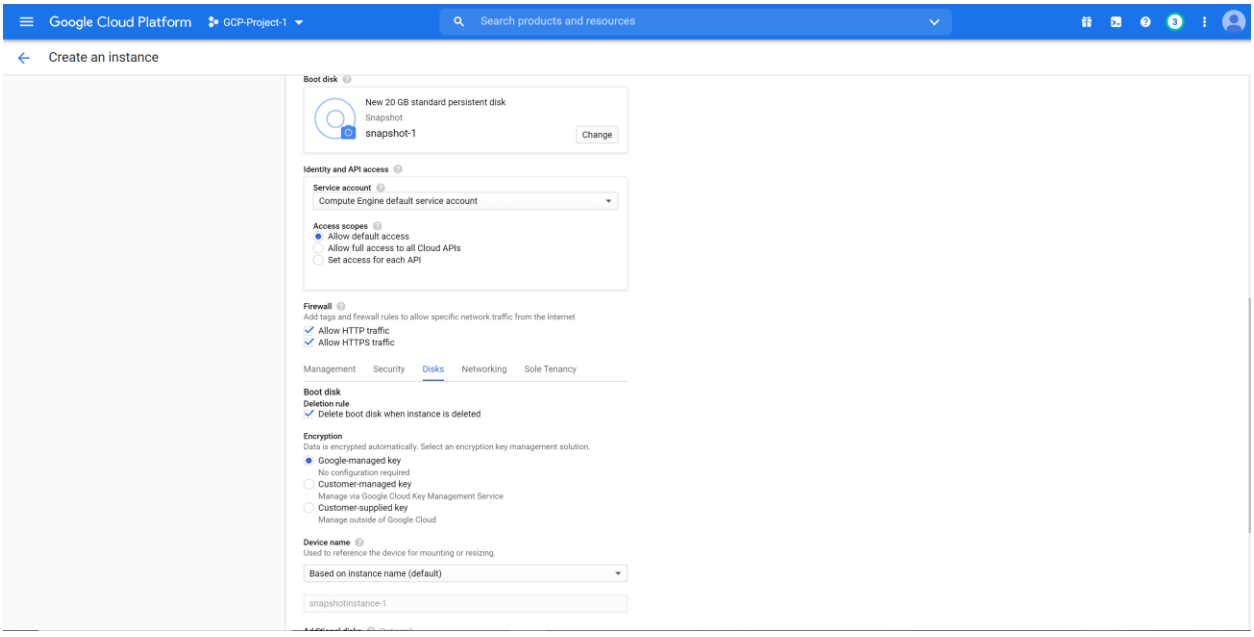
Created a snapshot of the public instance image to be used as a source for creating a new VM. Snapshot are of incremental nature i.e. snapshot taken at the very 1st time takes the whole point in time snapshot of the instance persistent disk and next time the snapshot is created only the modified data since the last snapshot would be captured. This incremental nature of snapshot makes them faster and easy to transport to restore the data as only the modified data is captured instead of taking the whole image of the instance every time thereby reducing the size of the snapshot.



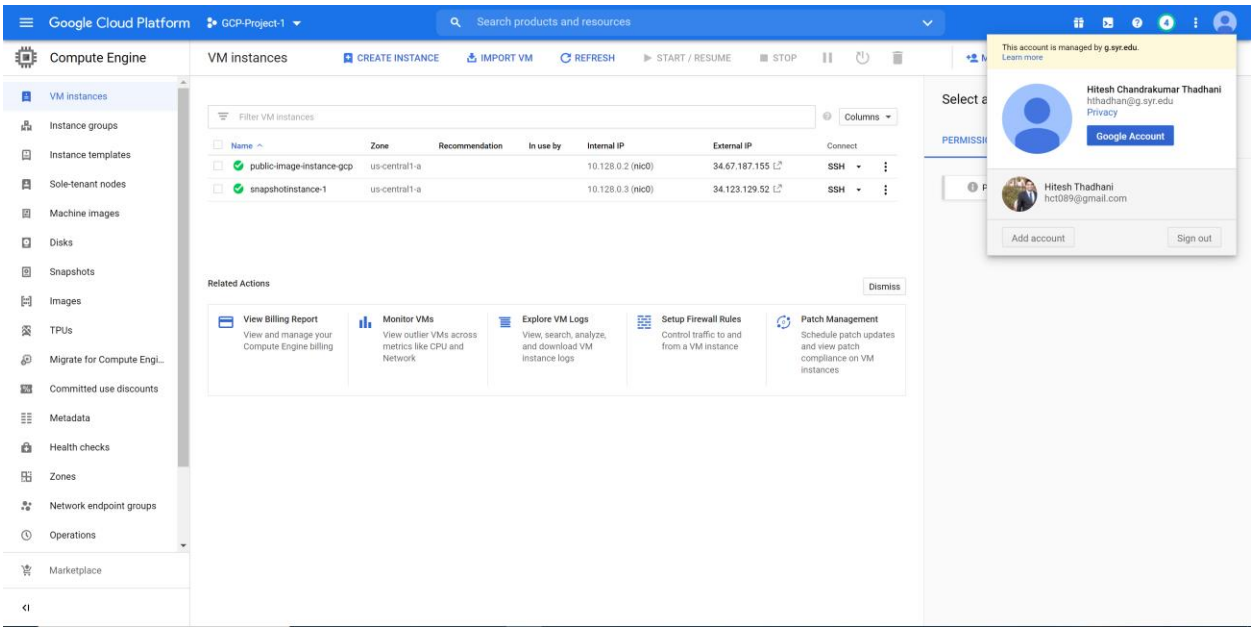
ii. Creating a new Instance from the Snapshot taken earlier

A new instance is created by using the snapshot as the boot disk taken earlier of the public image instance.





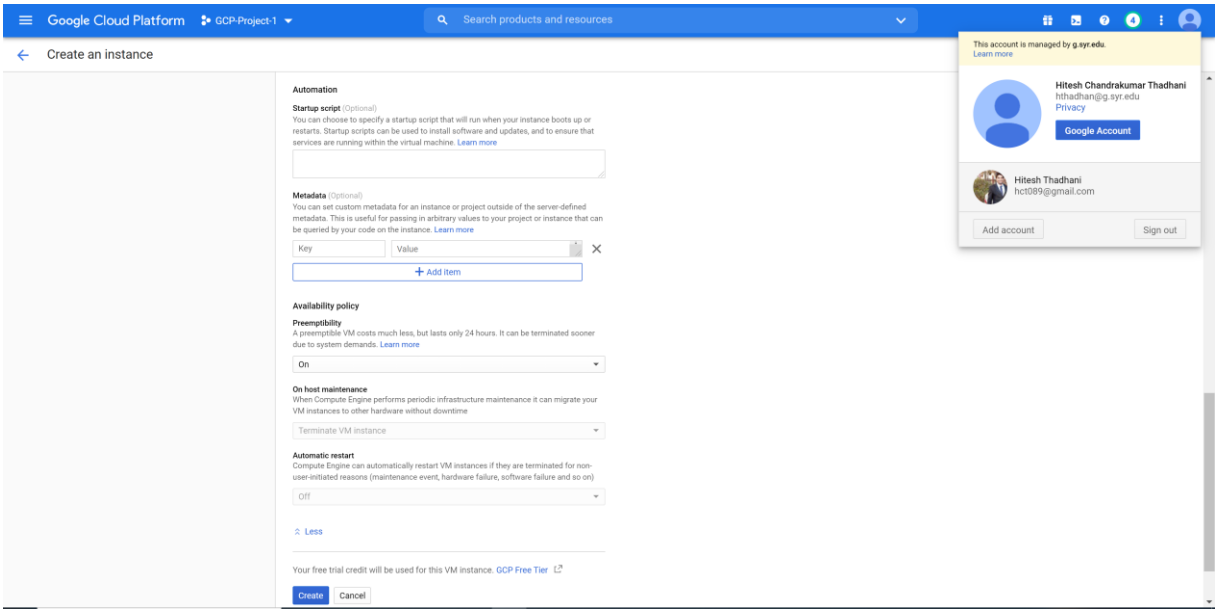
Allow HHTP and HTTPS traffic makes the VM use internet for the access allowing the traffic to be directed to the VM over the HTTPS and HTTP protocol. Delete the disk when instance is deleted allows to release the disk space whenever the instance is deleted.



C. Complete the following sub-modules in ***Creating and Starting a Preemptible VM Instance***

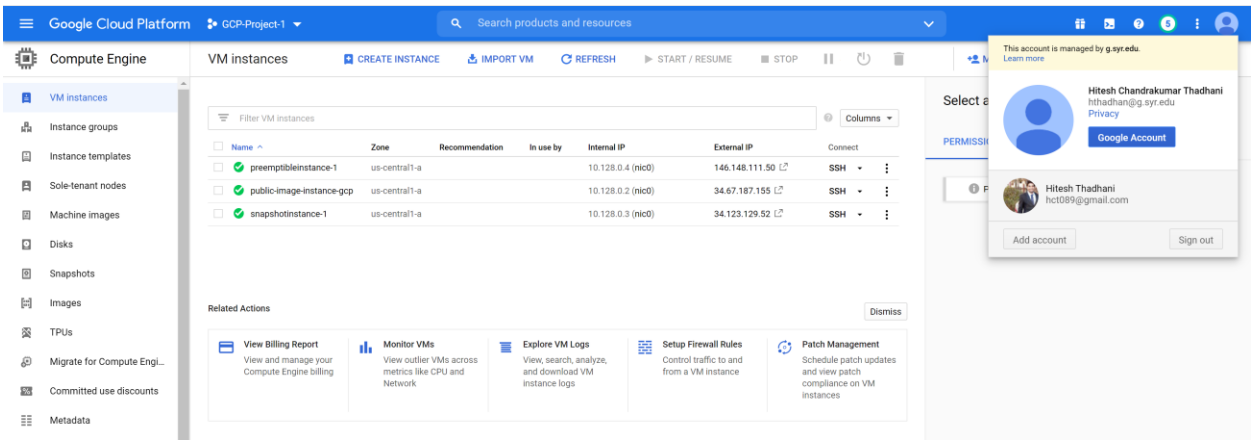
i. Creating a preemptible instance

Preemptible instance is a temporary instance which is available only for 24 hours to use at a very low price as compared to normal instances. These instances are not covered under any SLA. The compute resources can preempt or stop at any time if the load to the normal instances increases and it needs more compute power. These instances are useful for some batch processing tasks which are not so critical that if the instance is pre-empted there would be a business impact.



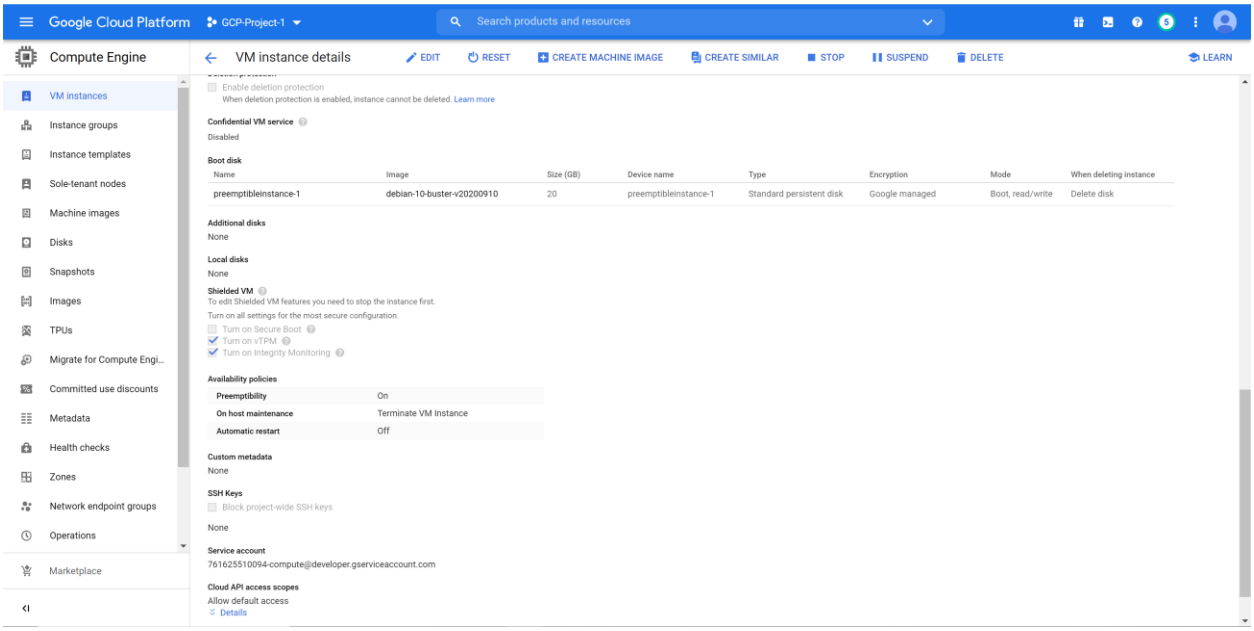
When the Preemptibility setting is turned on to create the preemptible instance, automatic restart of the instance is turned off and sets the On Host maintenance to Terminate VM instance.

Preemptible instance is successfully created and up and running as shown below.



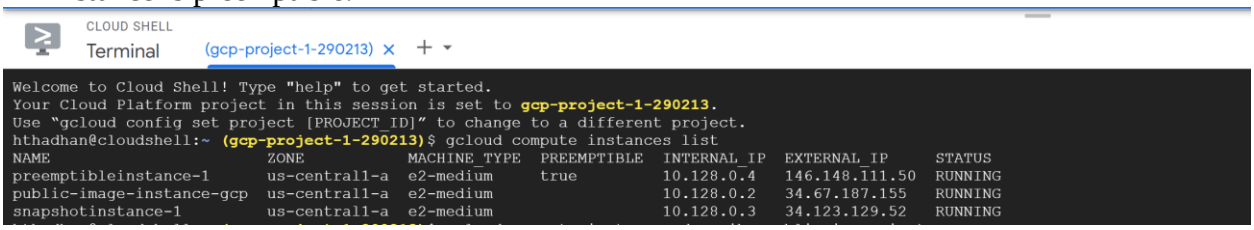
ii. Checking if an instance is preemptible

The instance is preemptible can be checked using 2 ways Console and Gcloud command line shell.

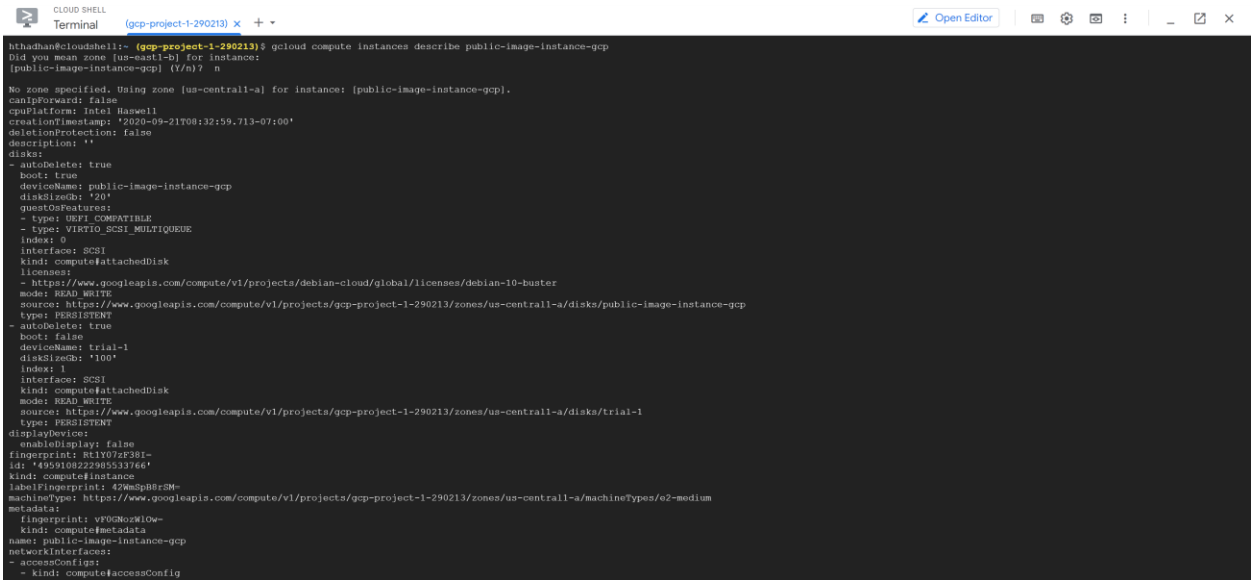


To check for the instance type if it is a preemptible instance or not, clicked on the instance and in the VM details, Availability policies shows that **Preemptibility is On** meaning that the instance is preemptible using the Console.

D. Check status of any instance you created earlier (Check any instance that you created earlier)
$ gcloud compute instances list – To list all the instances within a project. Also, for the preemptible instance created above, it shows PREEMPTIBLE to be true to show that this instance is preemptible.



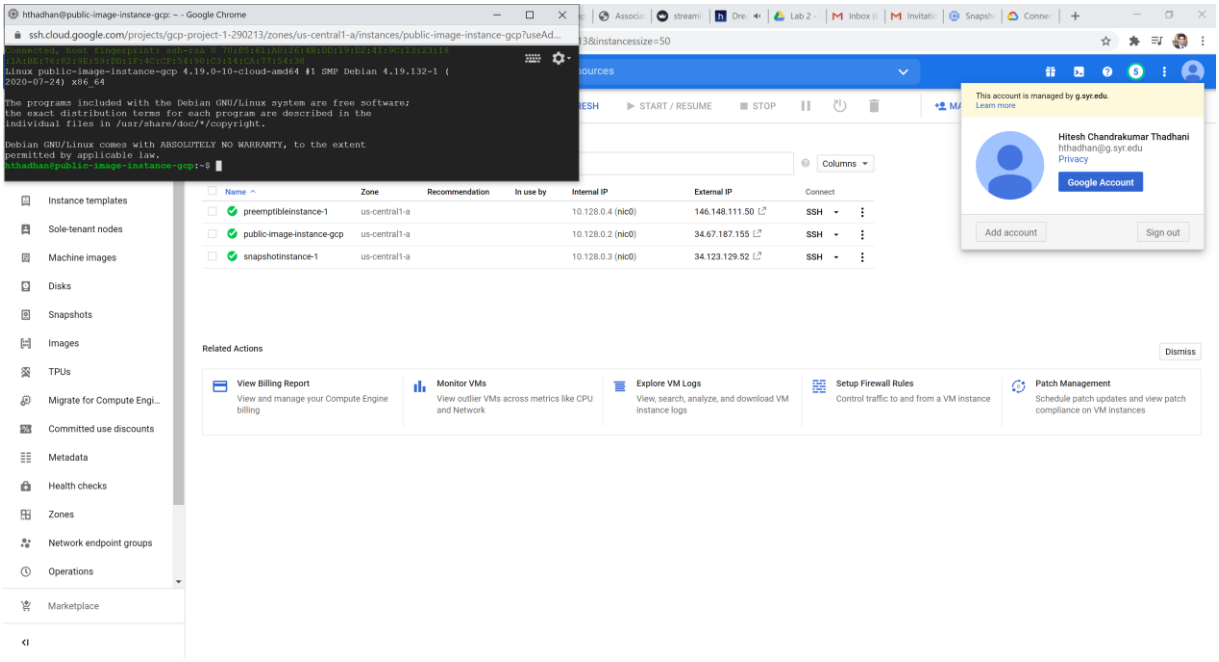
$ gcloud compute instances describe public-image-instance-gcp – This command describes a specific instance in a very detail giving all the information about the particular instance such as disksize, mode like Read/Write, network interface details like the External Public IP, Service Account details etc along with displaying the instance status if it is running, terminated, stopped etc.



E. Complete the following sub-modules in ***Connecting to instances***:

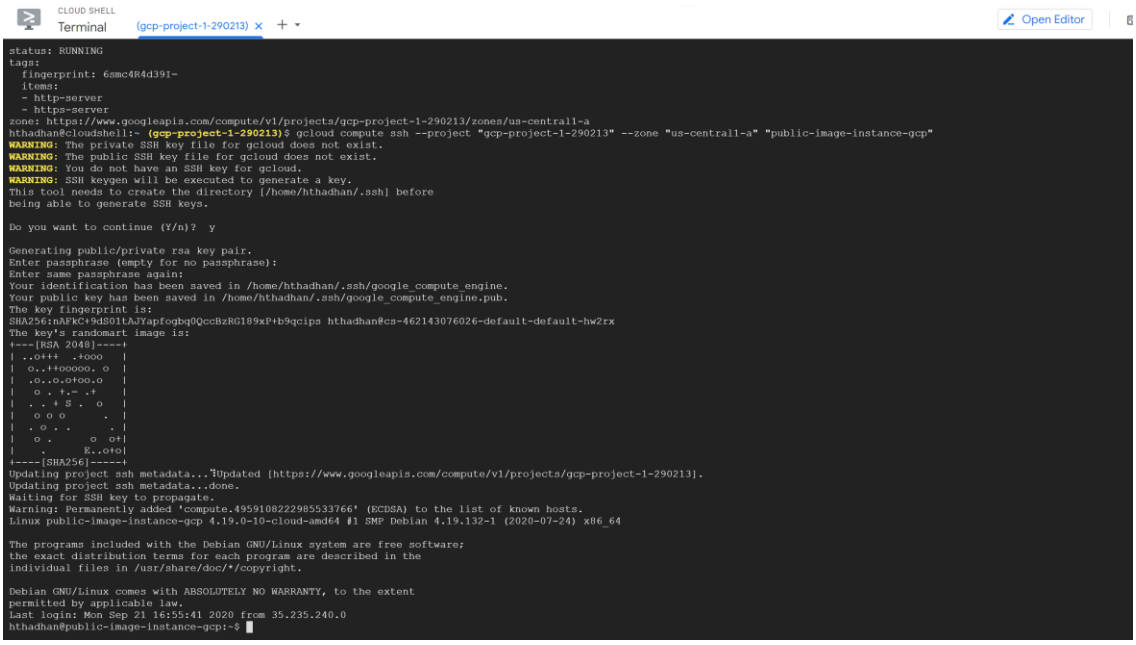
1. Connecting to Linux instances
    i. The public image VM instance is accessed through the use of ssh using browser

By clicking on ssh option next to the instance, a new SSH connection is created on the port 22 which is by default the port on which ssh runs. This creates a secure connection to the instance. When we connect to the instance using the ssh in a new browser window, it automatically transfers the required keys to make the secured connection to the instance.

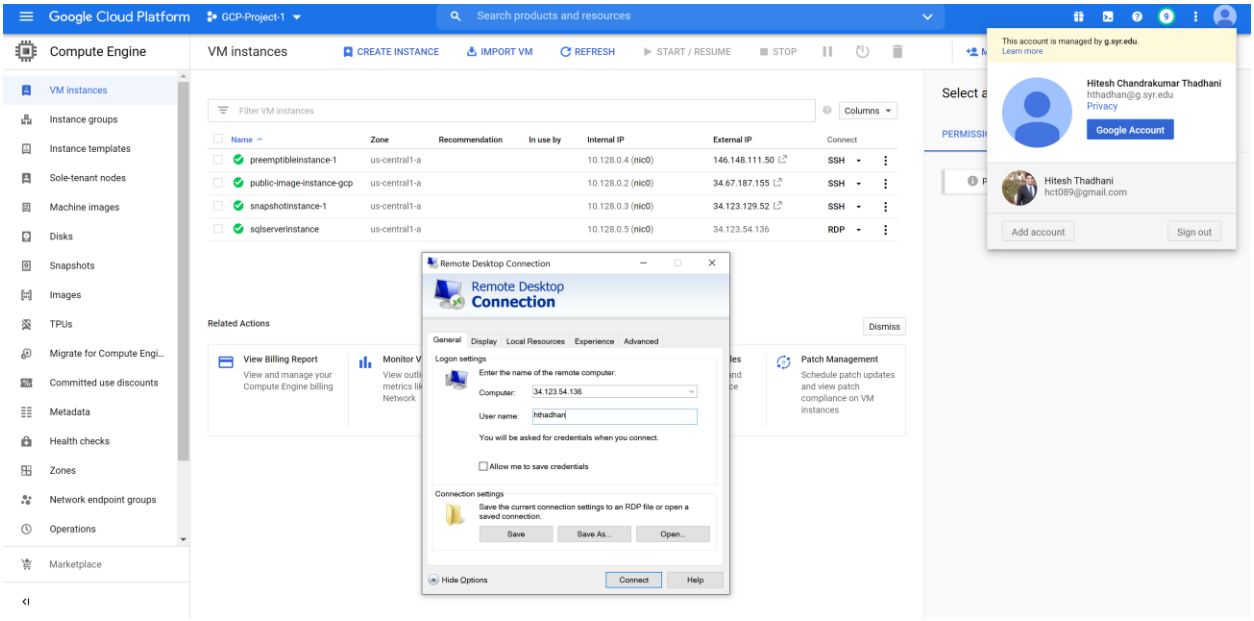


ii. The public image VM instance is being accessed by using the gcloud command-line tool

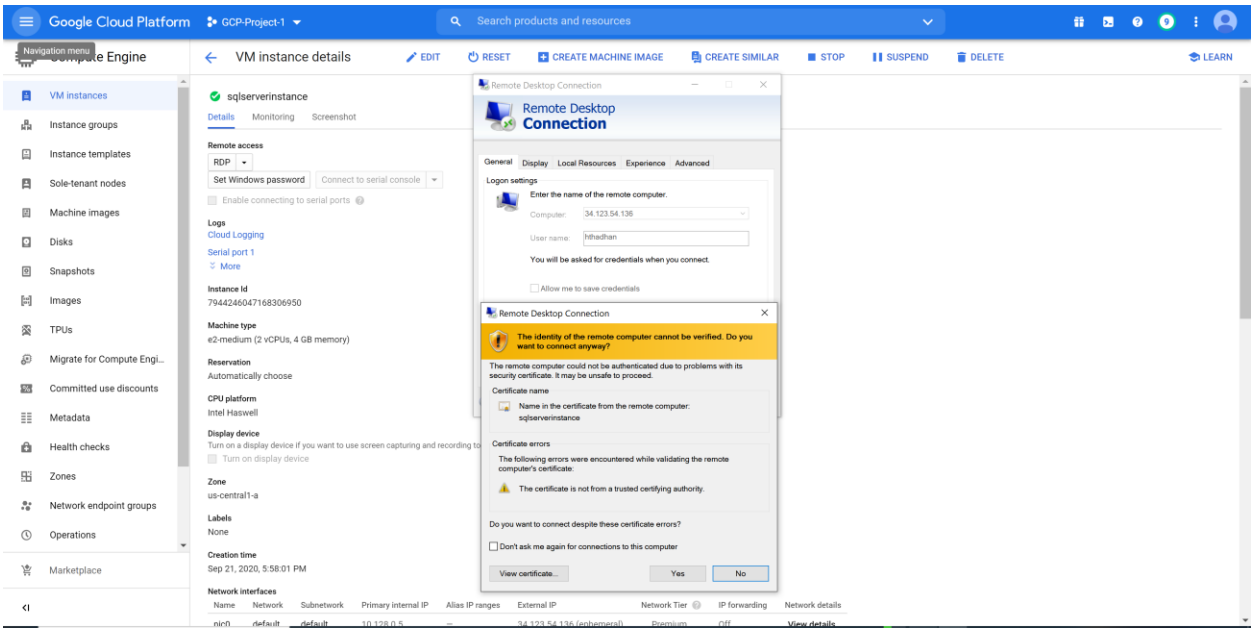
When connecting to the instance using the ssh using the Gcloud command line, public/private key pairs are generated, the project metadata is updated and these key pairs are used to authenticate the connection request to the instance.

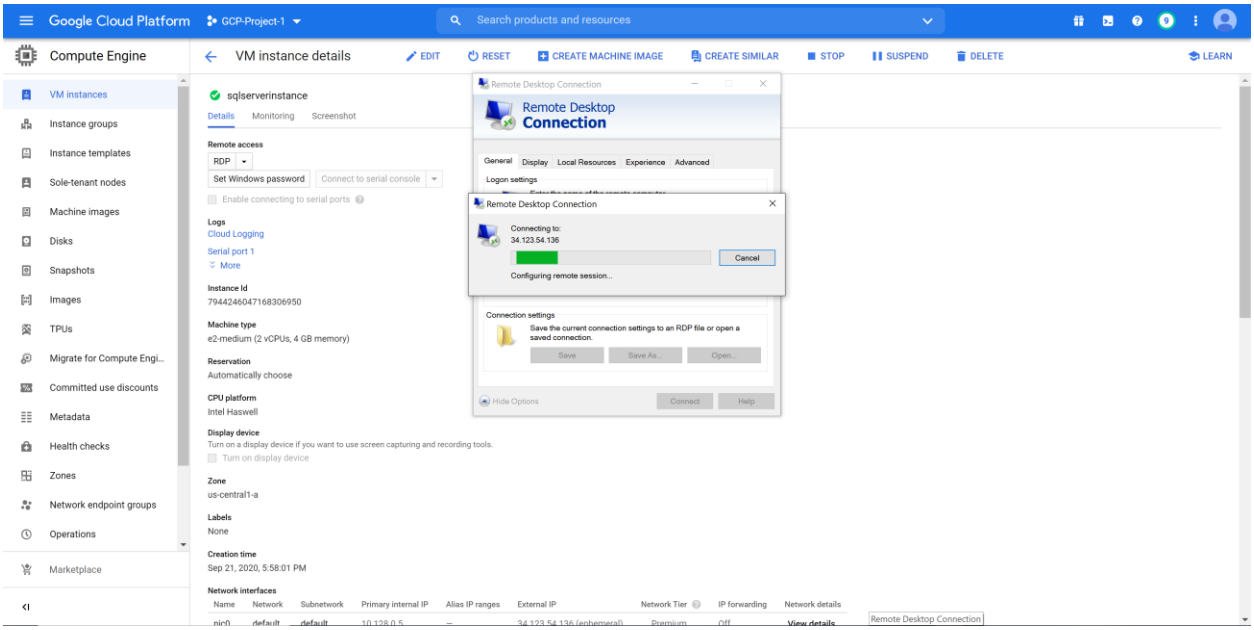


2. Connecting to Windows Instances

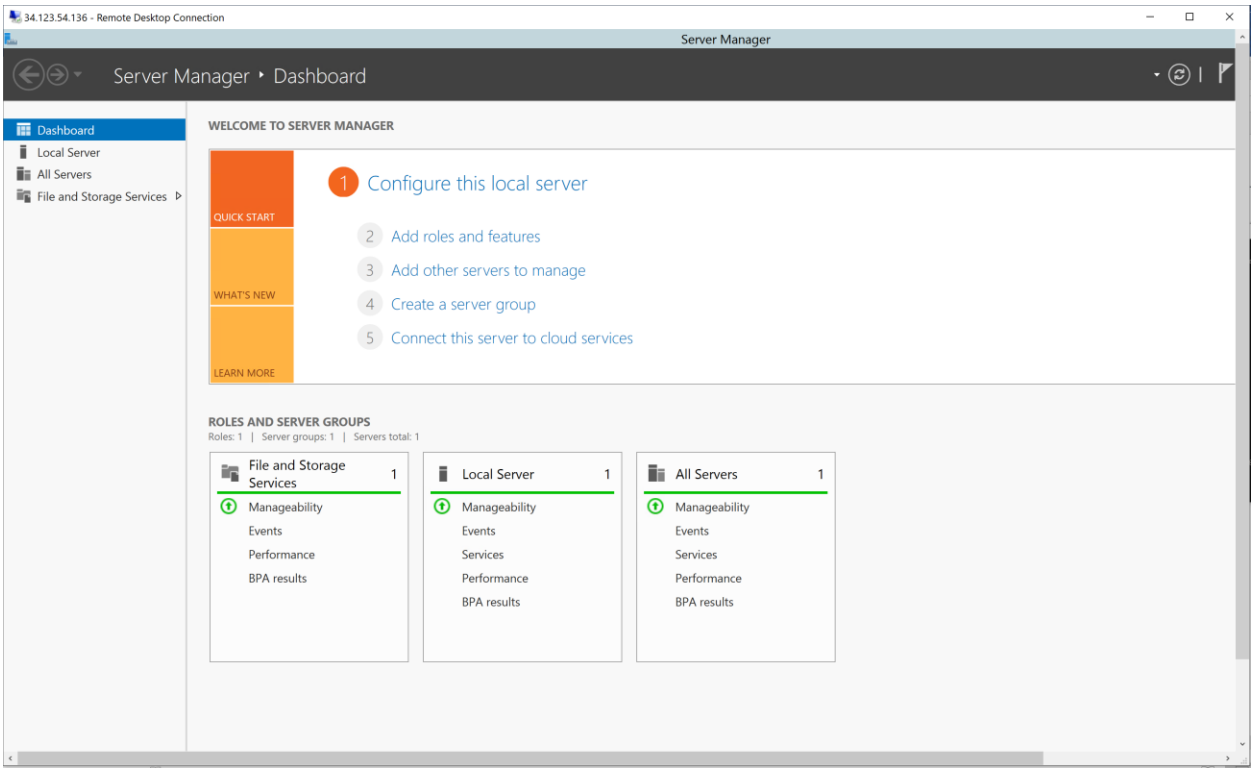


For connecting to the Windows instance, SQL server VM instance was created named sqlserverinstance. To connect to the windows instance with a public IP address, RDP (Remote Desktop Protocol) can be used. RDP will be used to login to the instance using the TCP traffic enabled on the port 1433. hthadhan is the username used to connect to the windows instance. Any SQL server using the Windows Operating System can be connected using RDP.





Successful connection to the SQL server Windows instance is seen below using the RDP.

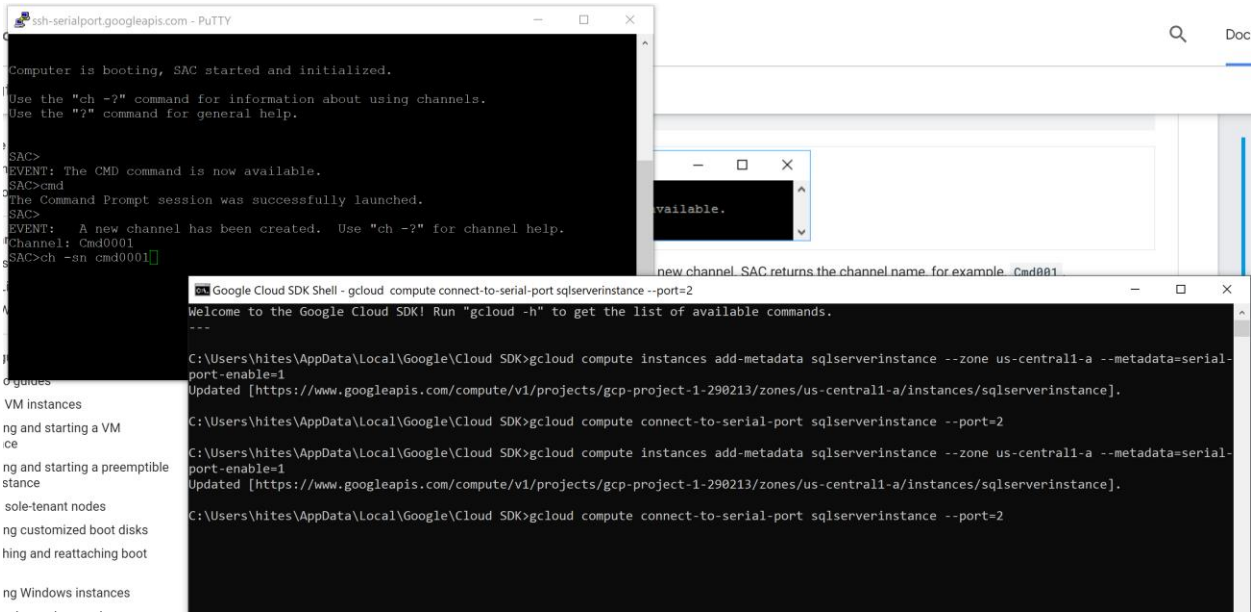


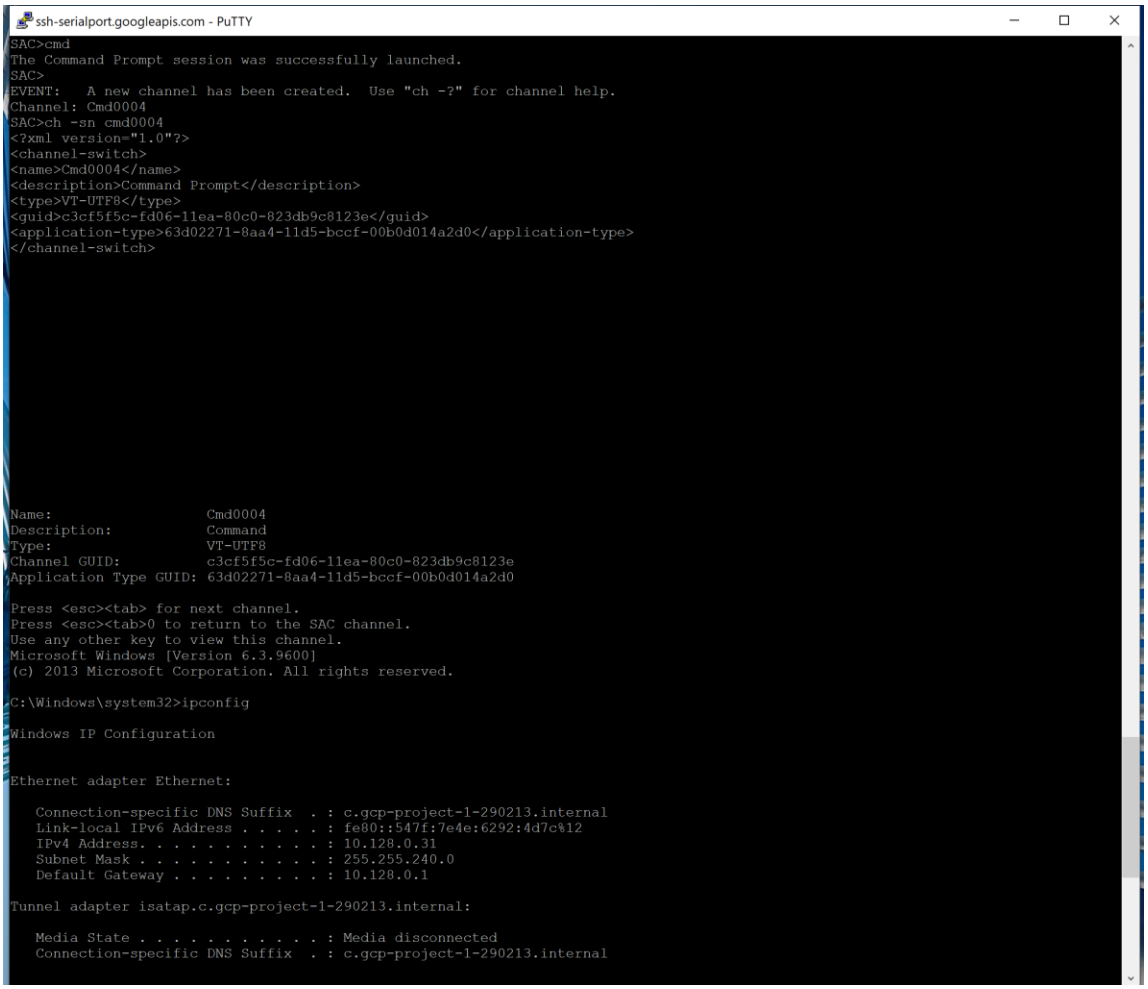
3. Special Administrative Console - Connecting to an instance through the command line
   (Connect to any instance that you created earlier)

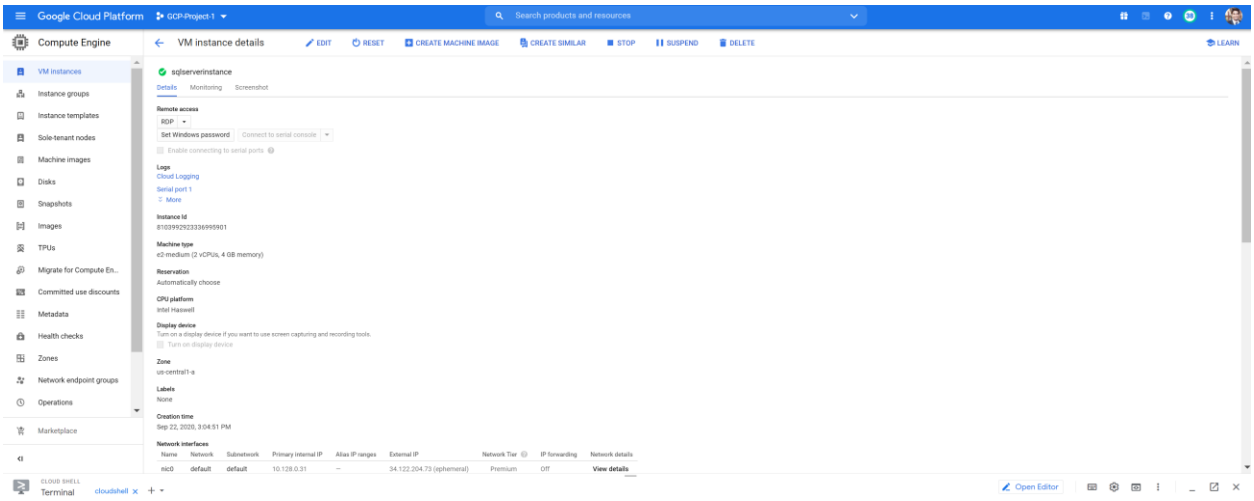
Connecting to the Windows Instance using the Special Administrative Console enables interactive interface to the Windows Server using the SAC> command prompt for managing Windows. This enables the connectivity to the Windows instance through a serial interactive console.

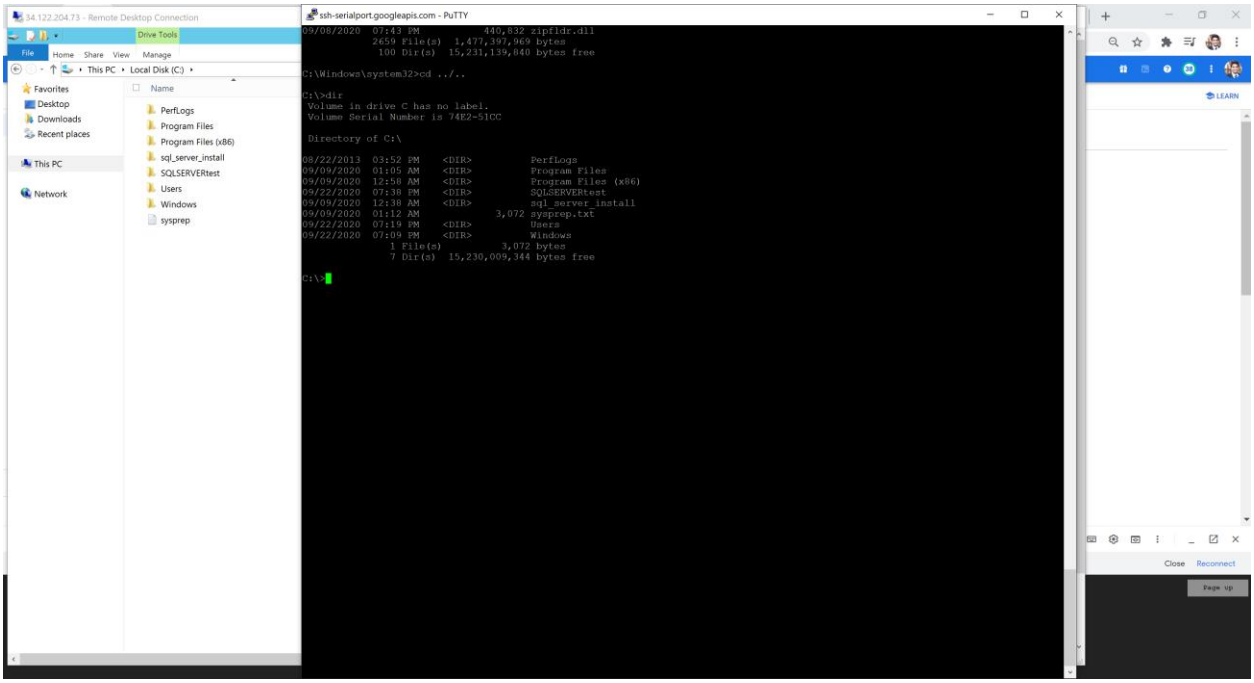
cmd - SAC command creates a new channel to be able to login to the Windows using the username and password.

Ch -sn cmd0001 – Switch the channel to the newly created channel to login to the Windows machine using this channel with a valid username and password.




ssh-serialport.googleapis.com - PuTTY
SAC>cmd
The Command Prompt session was successfully launched.
SAC>
EVENT: A new channel has been created. Use "ch -?" for channel help.
Channel: Cmd0004
SAC>ch -sn cmd0004
<?xml version="1.0"?>
<channel-switch>
<name>Cmd0004</name>
<description>Command Prompt</description>
<type>VT-UTF8</type>
<guid>c3cf5f5c-fd06-11ea-80c0-823db9c8123e</guid>
<application-type>63d02271-8aa4-11d5-bccf-00b0d014a2d0</application-type>
</channel-switch>
Name: Cmd0004
Description: Command
Type: VT-UTF8
Channel GUID: c3cf5f5c-fd06-11ea-80c0-823db9c8123e
Application Type GUID: 63d02271-8aa4-11d5-bccf-00b0d014a2d0
Press <esc><tab> for next channel.
Press <esc><tab>0 to return to the SAC channel.
Use any other key to view this channel.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Windows\system32>ipconfig
Windows IP Configuration
Ethernet adapter Ethernet:
Connection-specific DNS Suffix . : c.gcp-project-1-290213.internal
Link-local IPv6 Address . . . . . : fe80::547f:7e4e:6292:4d7c%12
IPv4 Address. . . . . : 10.128.0.31
Subnet Mask . . . . . : 255.255.240.0
Default Gateway . . . . . : 10.128.0.1
Tunnel adapter isatap.c.gcp-project-1-290213.internal:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . : c.gcp-project-1-290213.internal


Google Cloud Platform
Compute Engine
VM instance details
EDIT
RESET
CREATE MACHINE IMAGE
CREATE SIMILAR
STOP
SUSPEND
DELETE
sqlserverinstance
Details
Monitoring
Screenshot
Terminal



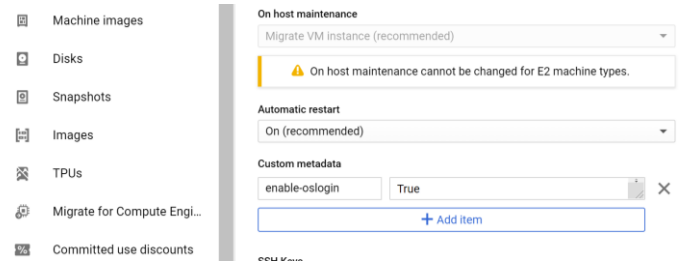
The SAC command prompt shows the directory structure of the windows OS instance using the interactive console.

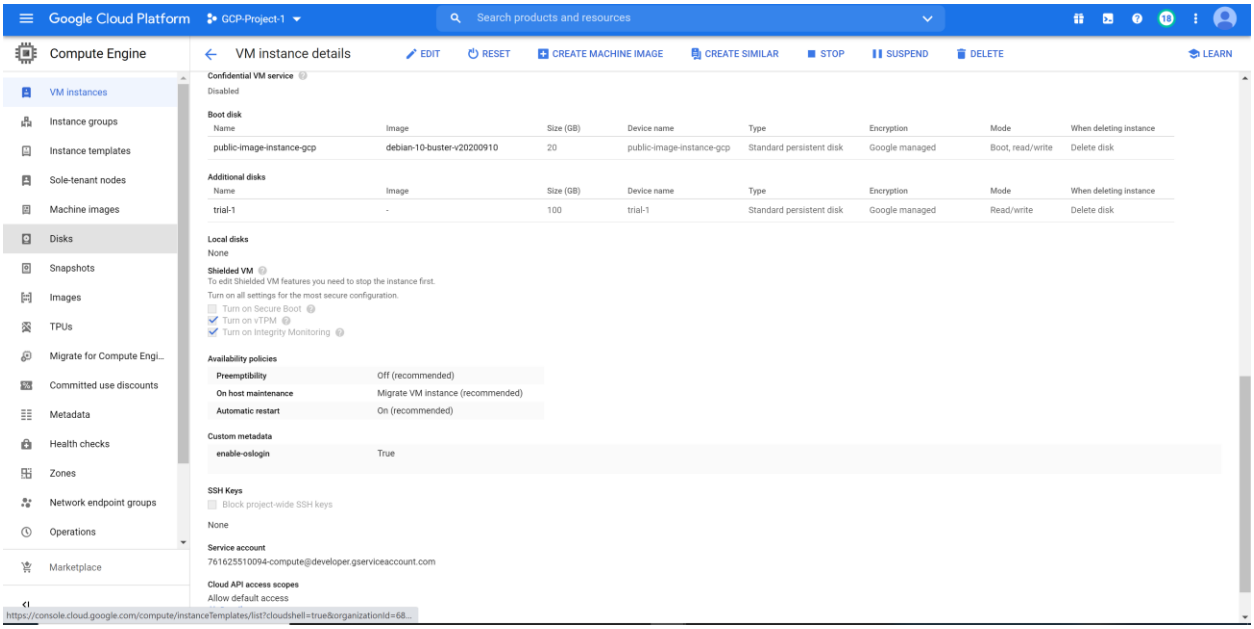
F. Complete the following sub-modules in ***Connecting to instances using advanced methods***:

Connecting to the instance requires authentication for the secure connection using ssh. This authentication would come by using public and private key pair used to make a secure connection.

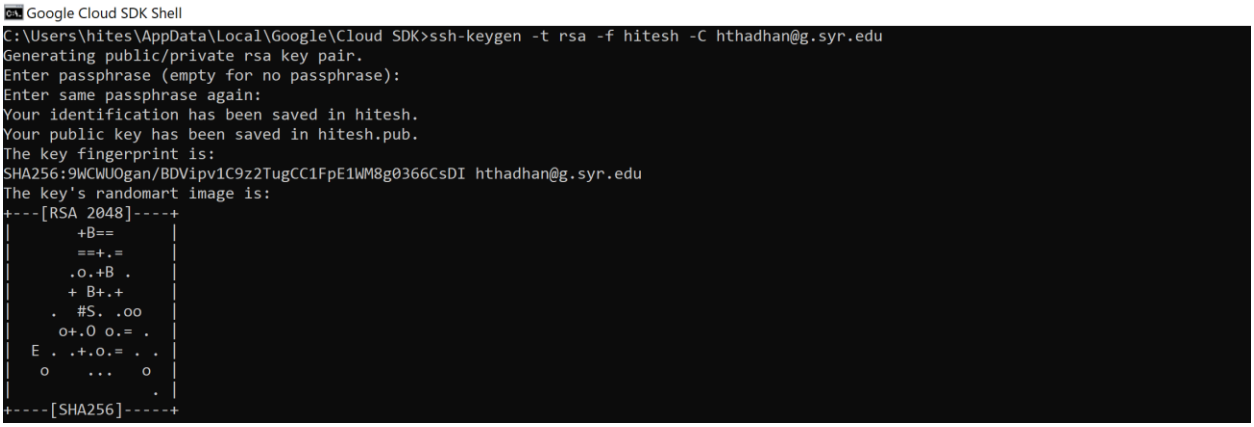
1. Providing public SSH keys to instances

To connect securely to the instance from outside, we need to get authenticated which is done by using public/private key pairs. Public and private keys are generated and public key is transferred to the VM instance. So, to provide the SSH key to the instance, we enabled OS Login for the instance which provides these keys through the Google Account to avoid hassle of managing the keys by us. The screenshot shows that the key pairs generated would be assigned to the instance as OS login is enabled.

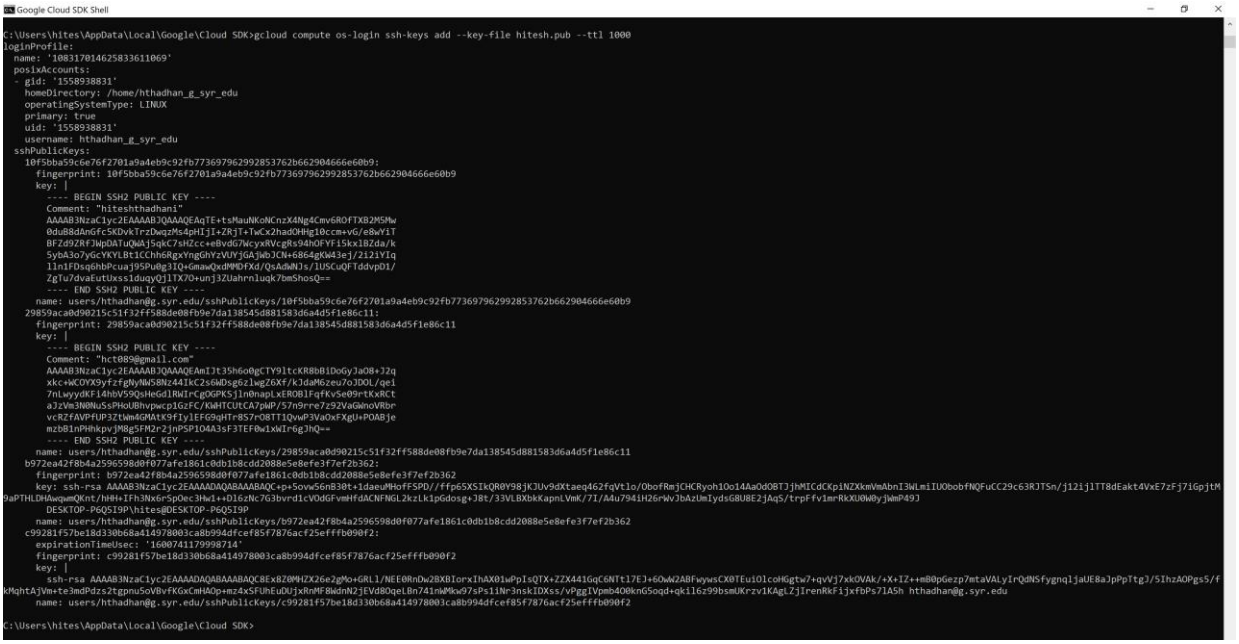




Now we create the key pairs which would be used to authenticate the login. The public/private key pairs are generated using the keygen of the type RSA. The file containing the public keys is stored as hitesh.pub and private key file is named as hitesh. The value after -C is the google account to which these keys would be provided to enable the authentication. These public/private key pairs are now stored locally.



The public keys are transferred to the instance for the instance to authenticate the login. The ttl is optional parameter which gives the time to live and is set to 1000 for the request to go the instance and live for some time before expiration.

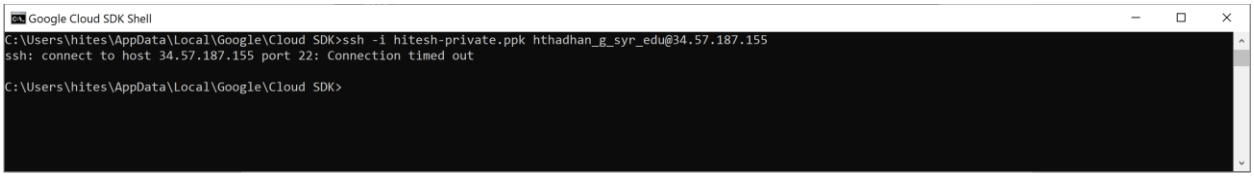


2. Connecting using third-party tools (SSH)

Now that the public keys are transferred to the VM instance for authentication, we use the private key to be able to allow the access as these are key pairs. The format to login to the instance is

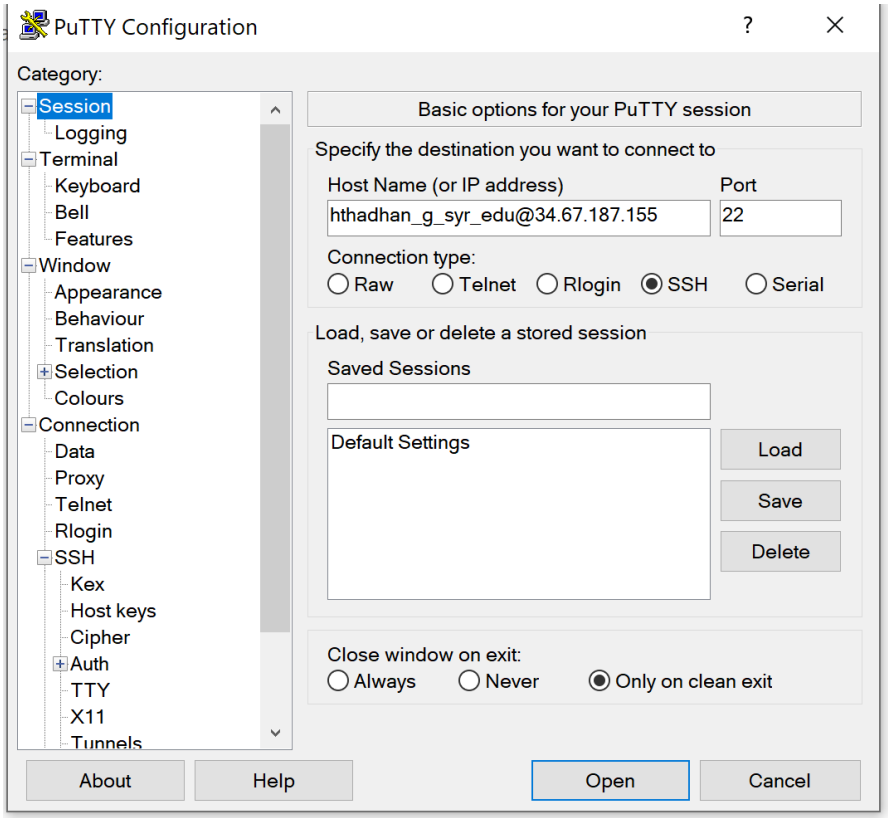
ssh -i [PRIVATE_KEY_PATH] [USERNAME]@[EXTERNAL_IP_ADDRESS]

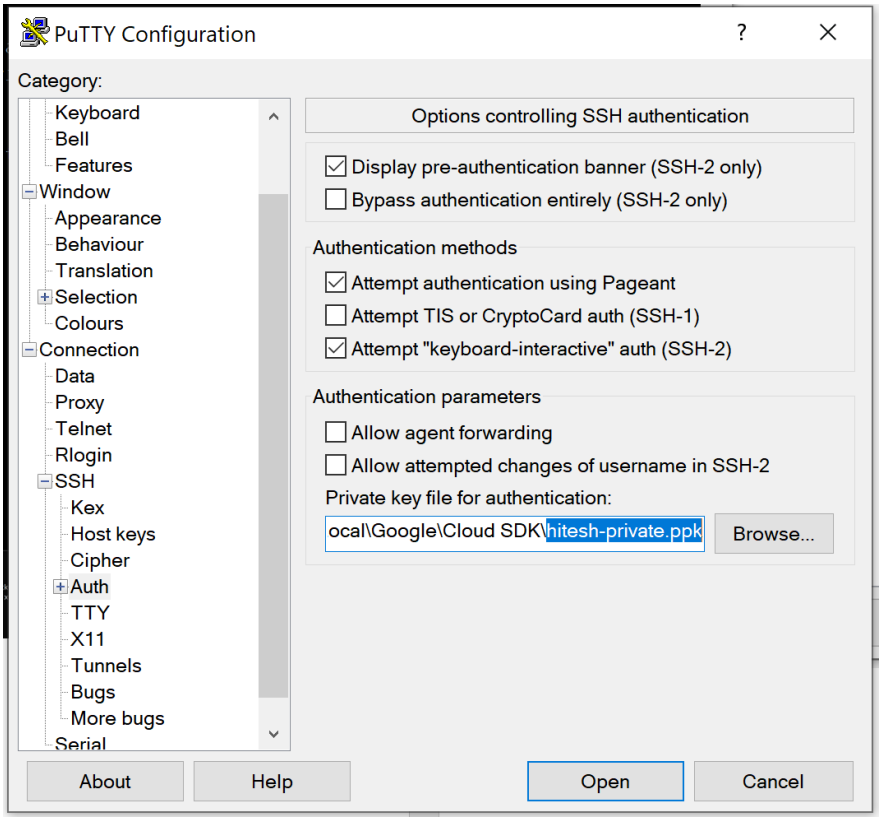
The authentication is successful and we do not have any error related to authentication like public key error only connection timed out. Successfully created secured ssh connection to the instance using the public/private key pairs transferred to the VM instance for the authentication.



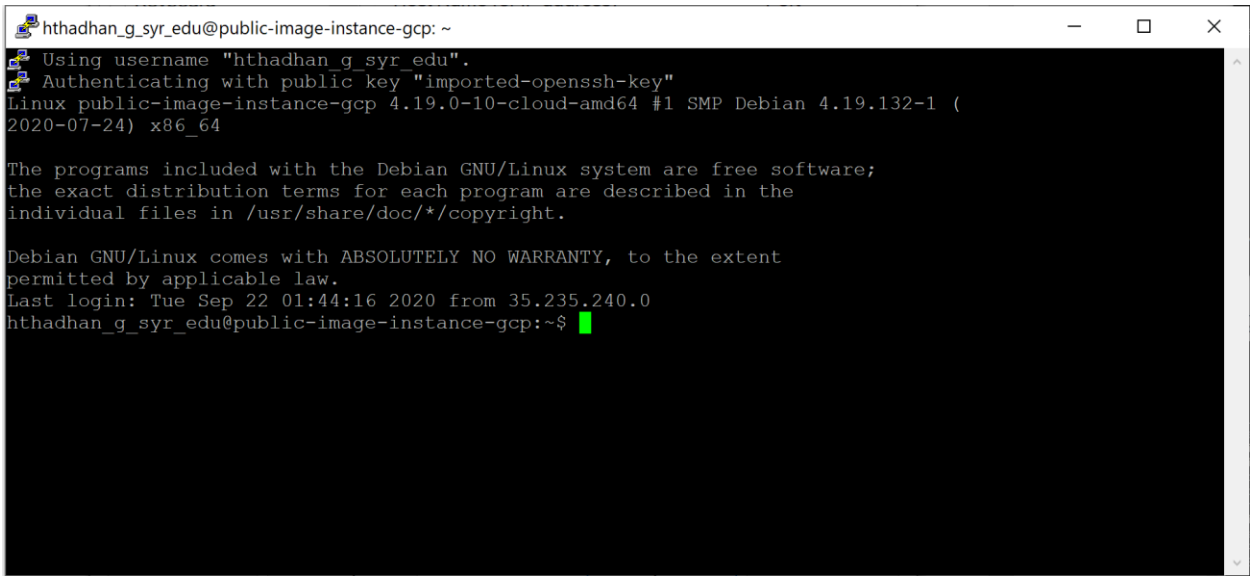
Below shows how we can use Putty to connect to the VM instance with the Username we mentioned while generating the public/private key pairs using keygen.

Private key path is provided in the putty at SSH -> Auth and username along with External IP address of the instance to login specified in the IP Address.


PuTTY Configuration
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
Telnet
Rlogin
SSH
Kex
Host keys
Cipher
Auth
TTY
X11
Tunnels
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address)
Port
hthadhan_g_syr_edu@34.67.187.155
22
Connection type:
Raw
Telnet
Rlogin
SSH
Serial
Load, save or delete a stored session
Saved Sessions
Default Settings
Load
Save
Delete
Close window on exit:
Always
Never
Only on clean exit
About
Help
Open
Cancel


PuTTY Configuration
Category:
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
Telnet
Rlogin
SSH
Kex
Host keys
Cipher
Auth
TTY
X11
Tunnels
Bugs
More bugs
Serial
Options controlling SSH authentication
Display pre-authentication banner (SSH-2 only)
Bypass authentication entirely (SSH-2 only)
Authentication methods
Attempt authentication using Pageant
Attempt TIS or CryptoCard auth (SSH-1)
Attempt "keyboard-interactive" auth (SSH-2)
Authentication parameters
Allow agent forwarding
Allow attempted changes of username in SSH-2
Private key file for authentication:
ocal\Google\Cloud SDK\hitesh-private.ppk
Browse...
About
Help
Open
Cancel

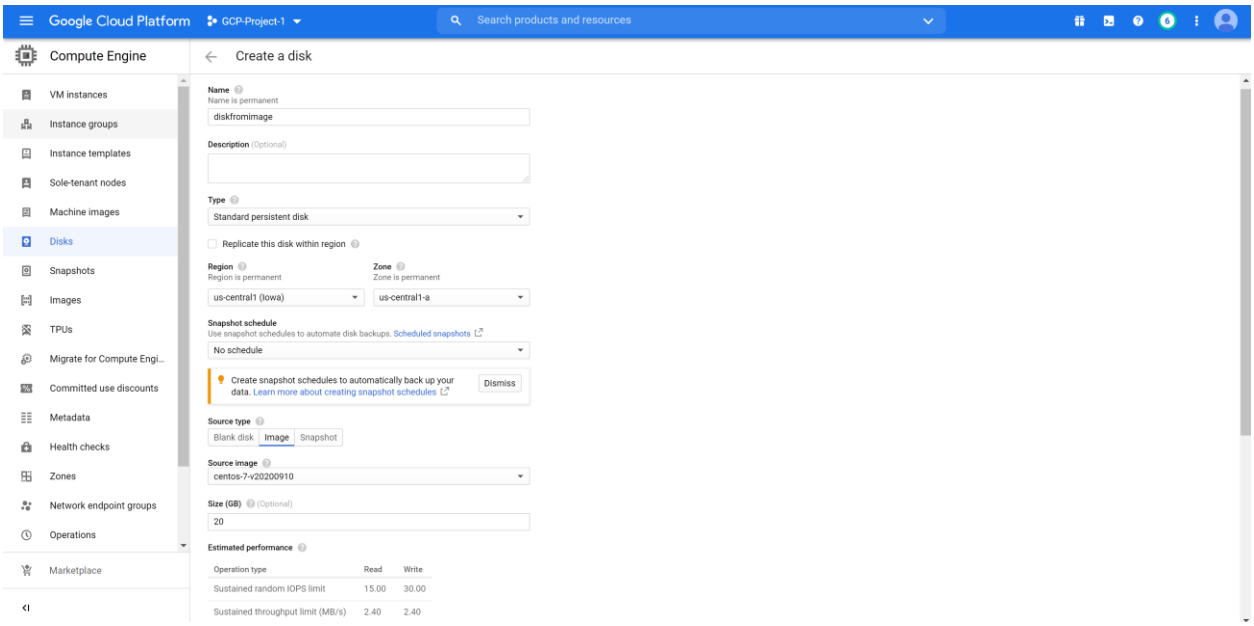
Successful authentication and login to the instance from the putty is shown below using the third-party tools (SSH) connection.



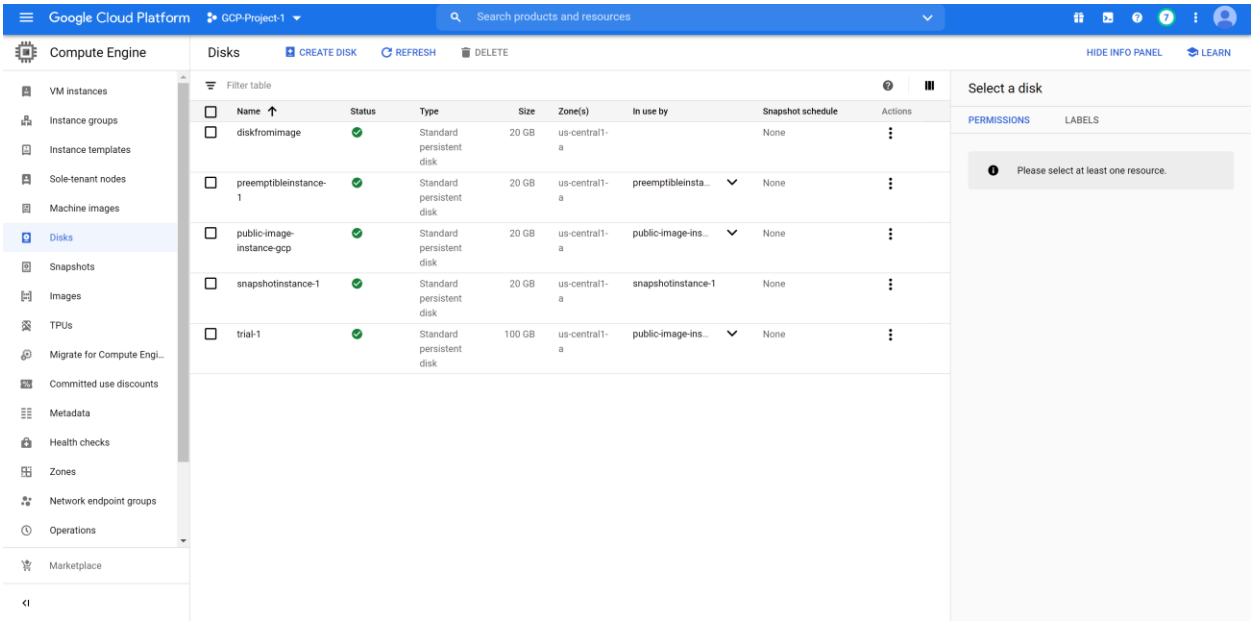
G. Complete the following sub-modules in ***Creating Customized Boot Disks***:

1. Creating a standalone boot persistent disk from an image:

A standalone disk is created with a size of 20GB which can be used later to be attached to any VM instance as an additional storage hard disk to store additional files or operating system files when the storage is almost full.

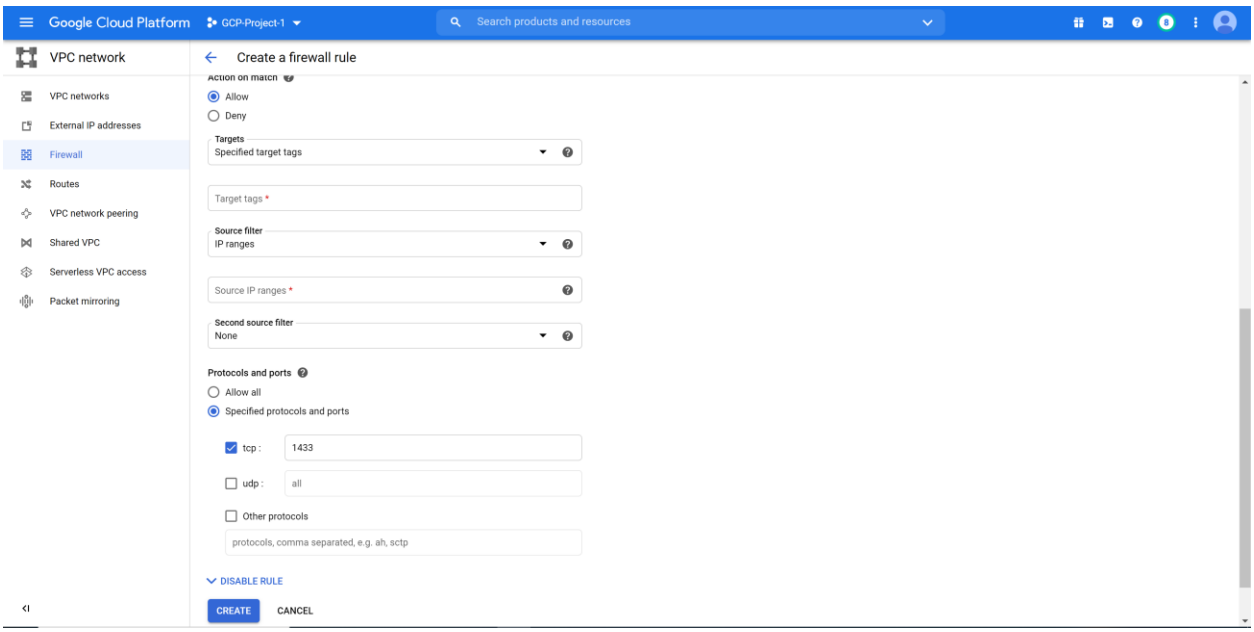


Standalone persistent boot disk creation is successful with a 20GB disk size is shown below.

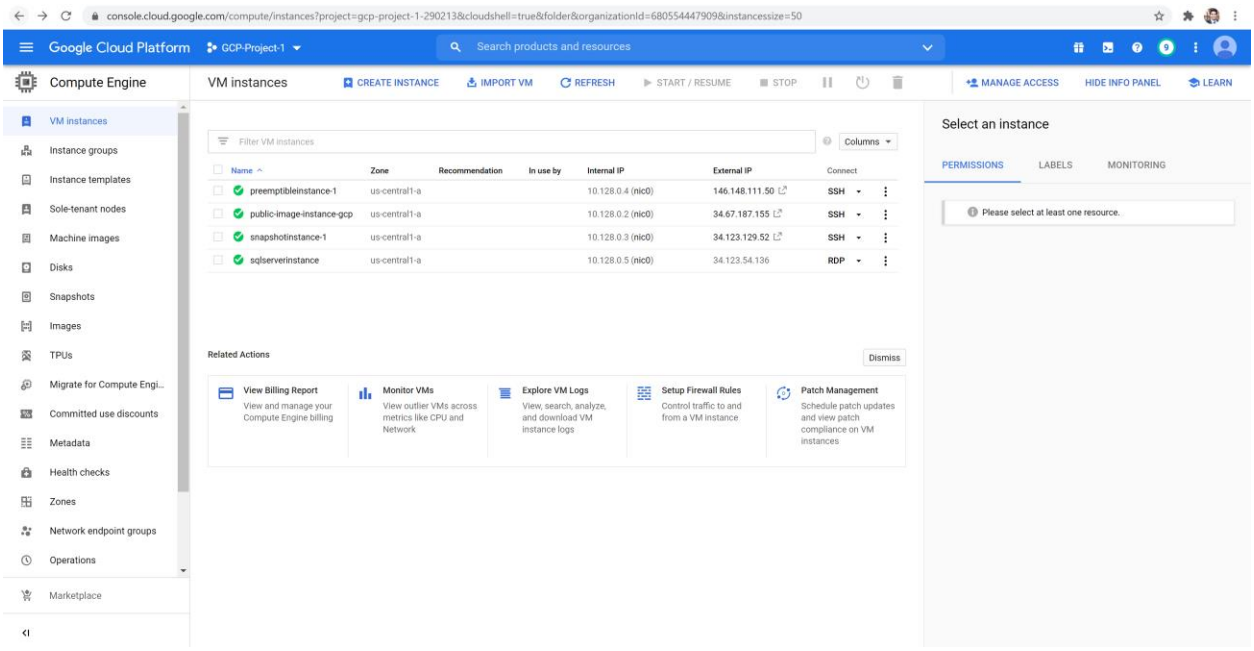


H. Creating an SQL Server Instance

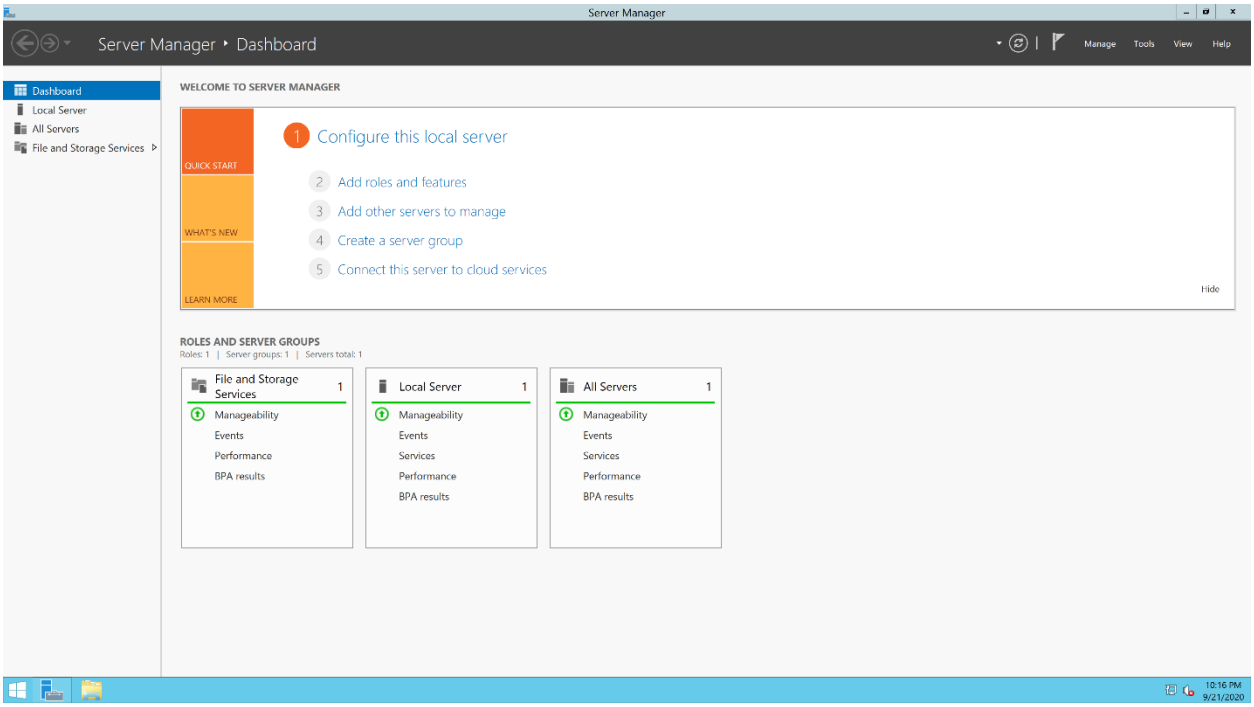
SQL Server instance is created using the SQL server 2012 Windows OS with a configuration for better performance like 2CPUS's etc. Firewall rule is created to enable the TCP traffic coming to the instance on the port 1433. This rule allows us to connect to the instance to be able to manage the SQL server database hosted on the Windows instance.

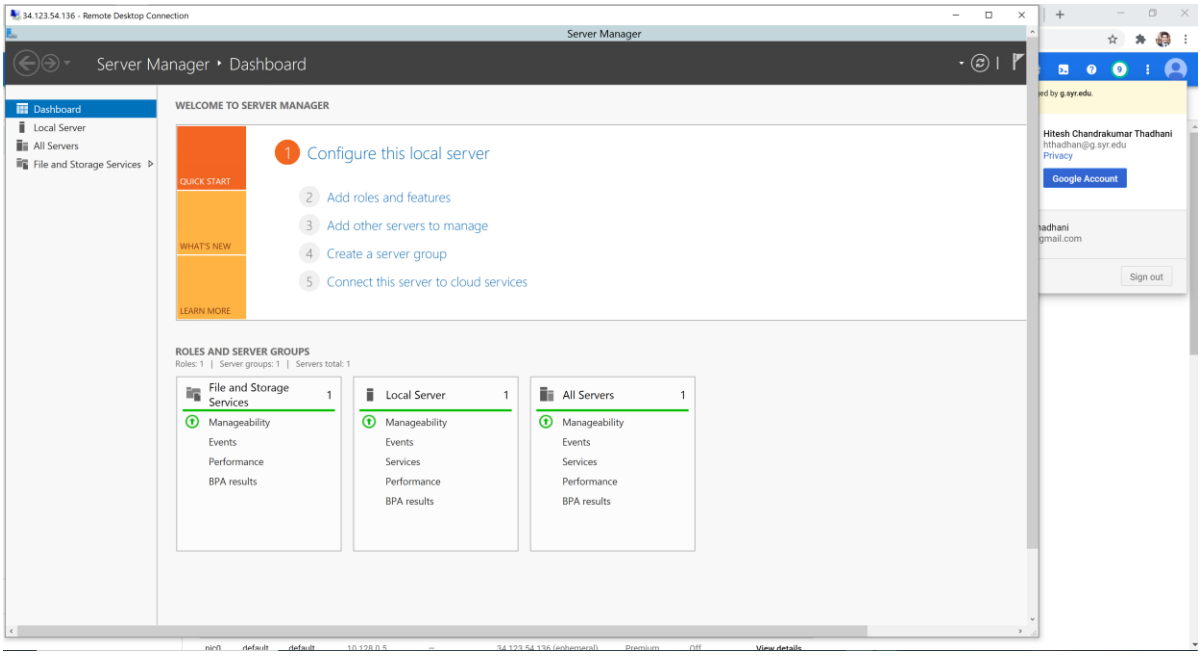


Successful creation of Windows instance is shown below.



The Windows instance with SQL server 2012 running is accessed using the RDP (Remote Desktop Protocol). RDP is used to connect to any Windows instance from outside to manage the instance and the database running on the Windows Machine.



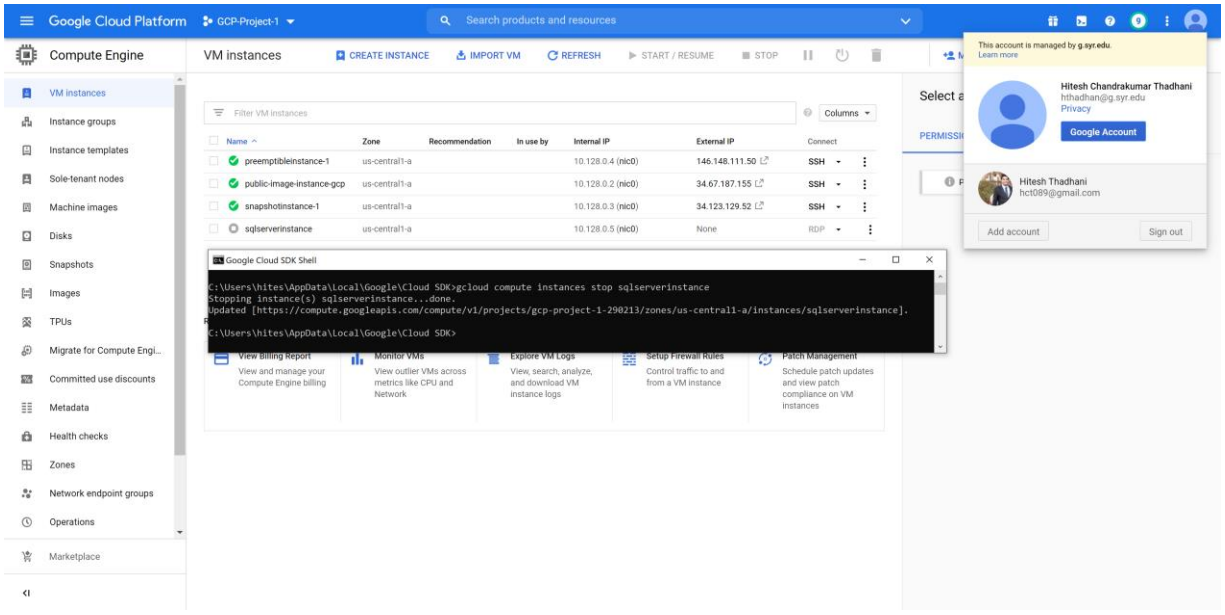


Section 5: Managing your instances

A. Stopping an Instance

1. Stopping an instance (GCloud)

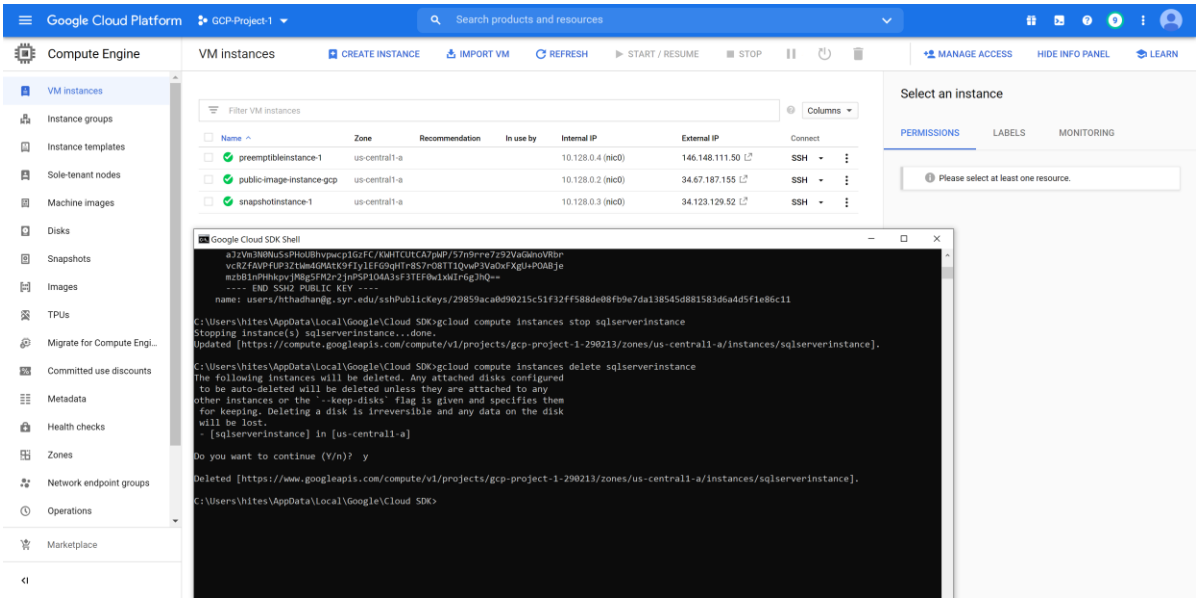
Stopping the sqlserverinstance using the Gcloud command line to save the cost when not using it.



B. Deleting an Instance
Complete the following sub-modules in ***Deleting an Instance***:

1. Delete an Instance (GCloud)

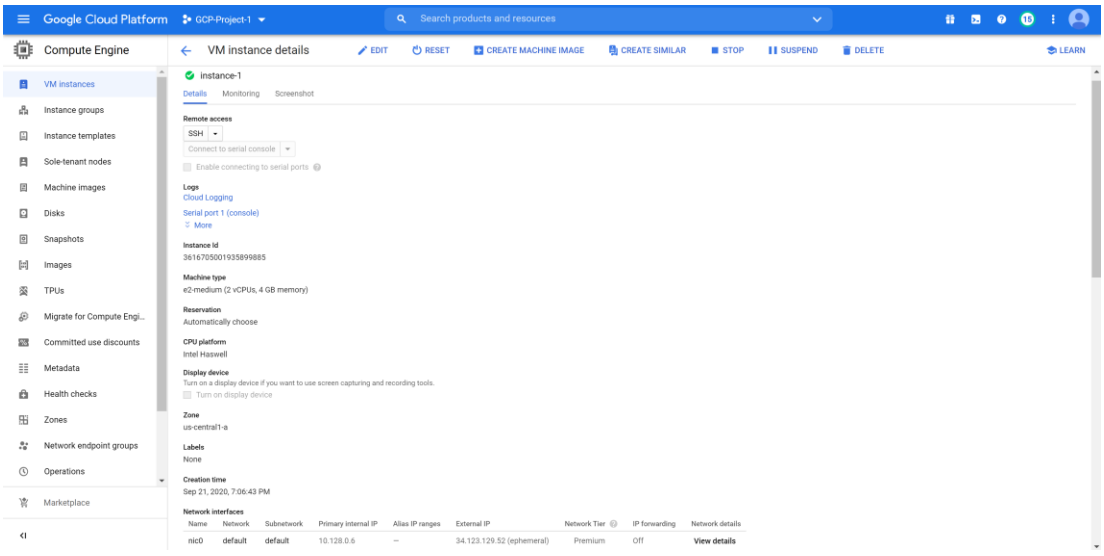
Deleting the sqlserverinstance using the Gcloud command line.

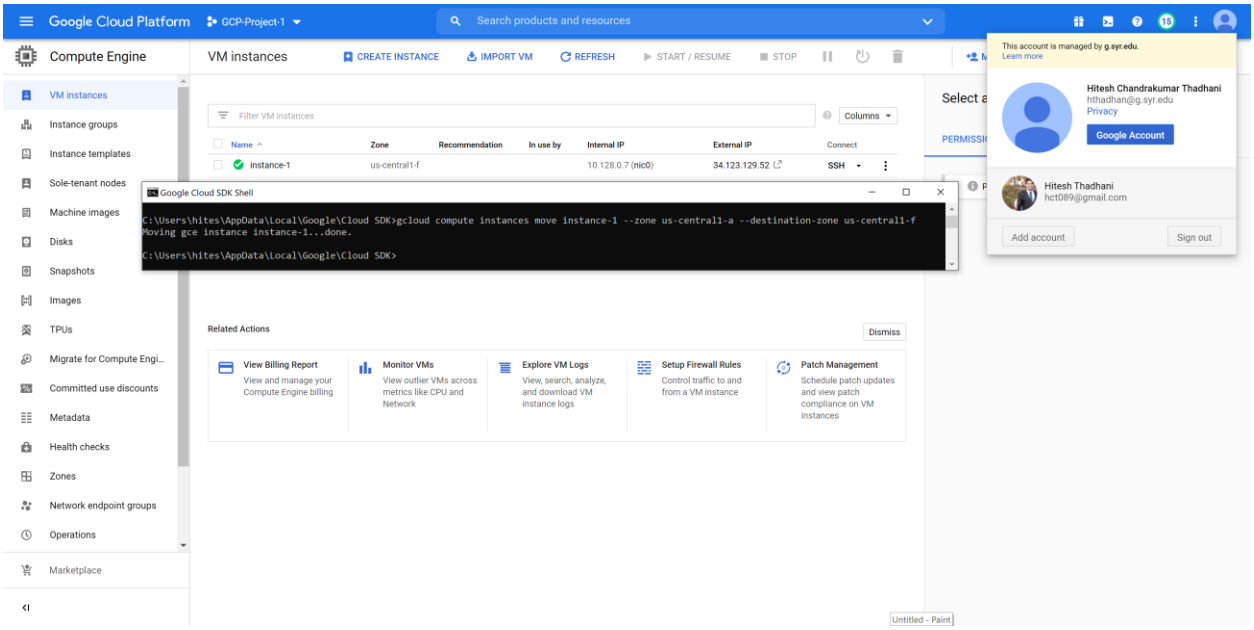


C. Moving an instance
Complete the following sub-module in ***Moving an instance between zones***:

1. Moving an instance automatically (GCloud)

The instance-1 which was in the Zone us-central1-a is being moved to us-central1-f using the Gcloud command line

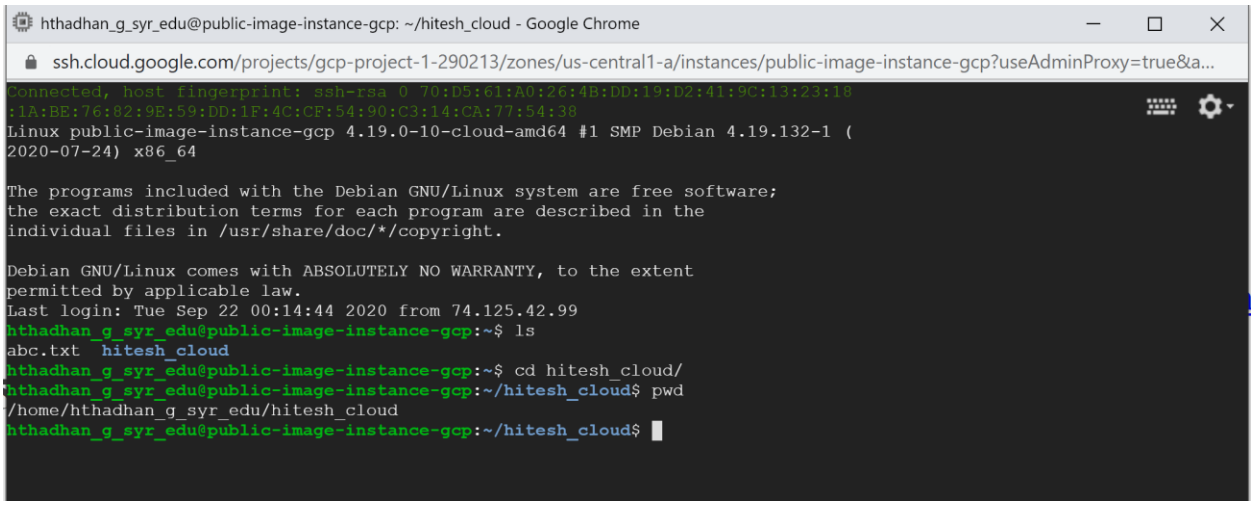




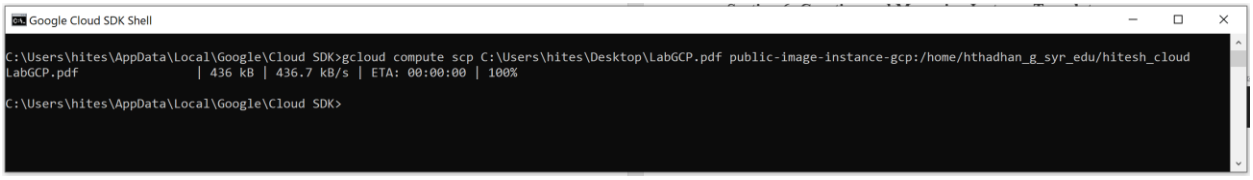
D. Complete the following sub-module in ***Performing Other Tasks With Your Instances***:
1. Copy files between an instance and local computer

Copying data from Local Machine to GCP Cloud Instance:

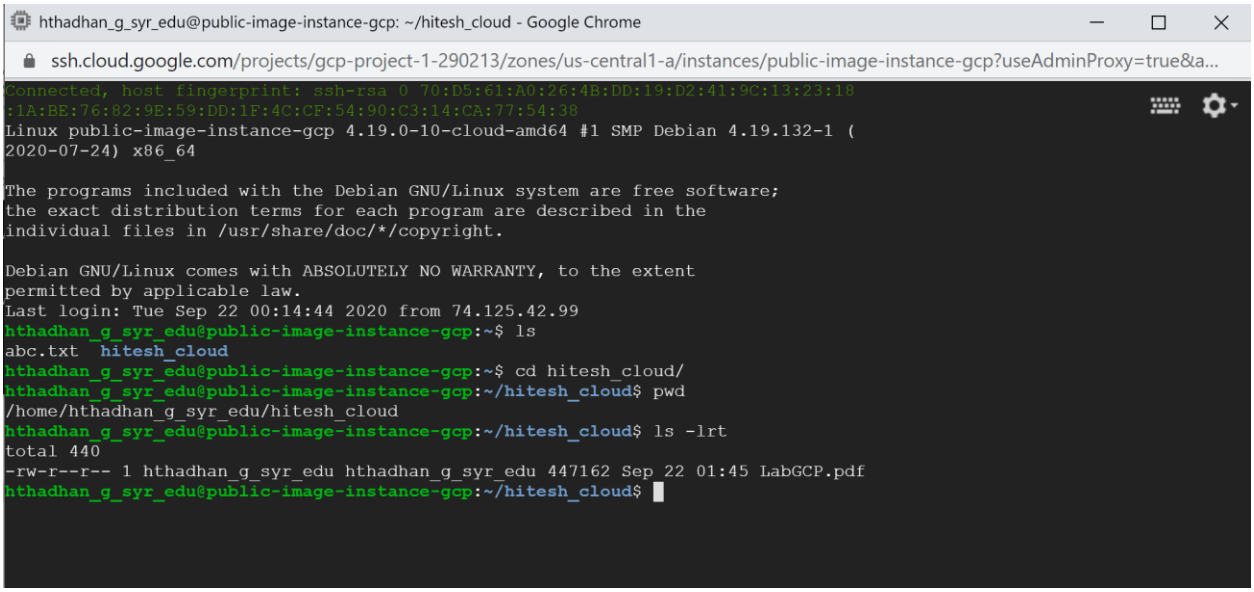
LabGCP pdf file was copied using gcloud scp command from the local machine to the instance. The below shows a new folder created with the name hitesh_cloud for the file to reside inside this folder.



The pdf file is copied successfully from the Local Machine to the hitesh_cloud folder in the VM as shown below.

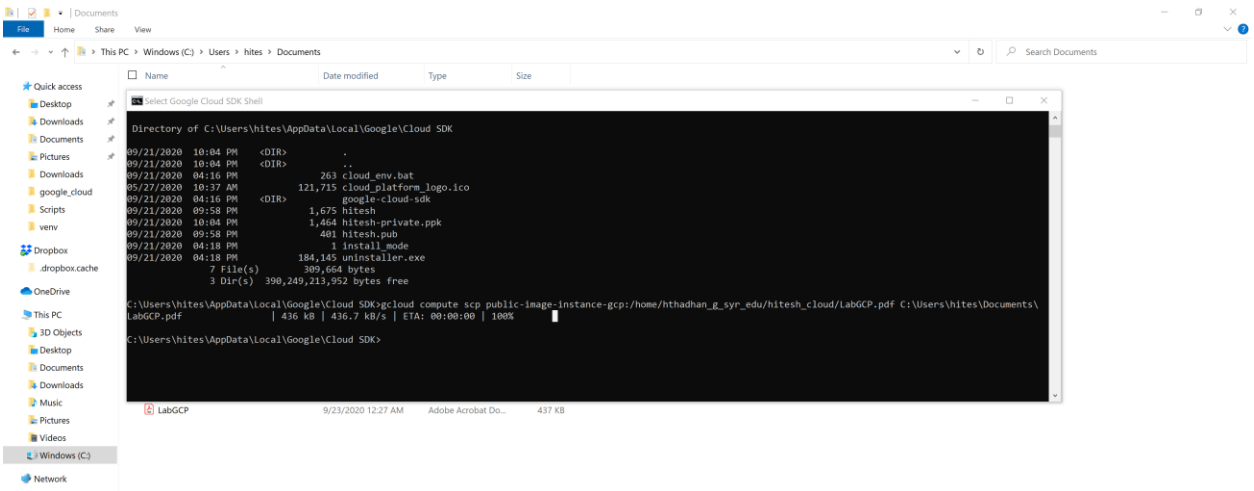


File is successfully residing inside the hitesh_cloud folder on the VM.



Copying data from GCP instance to Local Machine

The same file was copied from the VM instance to the Local Machine in the Documents folder which didn't have the file earlier. After the successful copy using the gcloud scp command, the file LabGCP.pdf is found in the Documents folder as shown below.

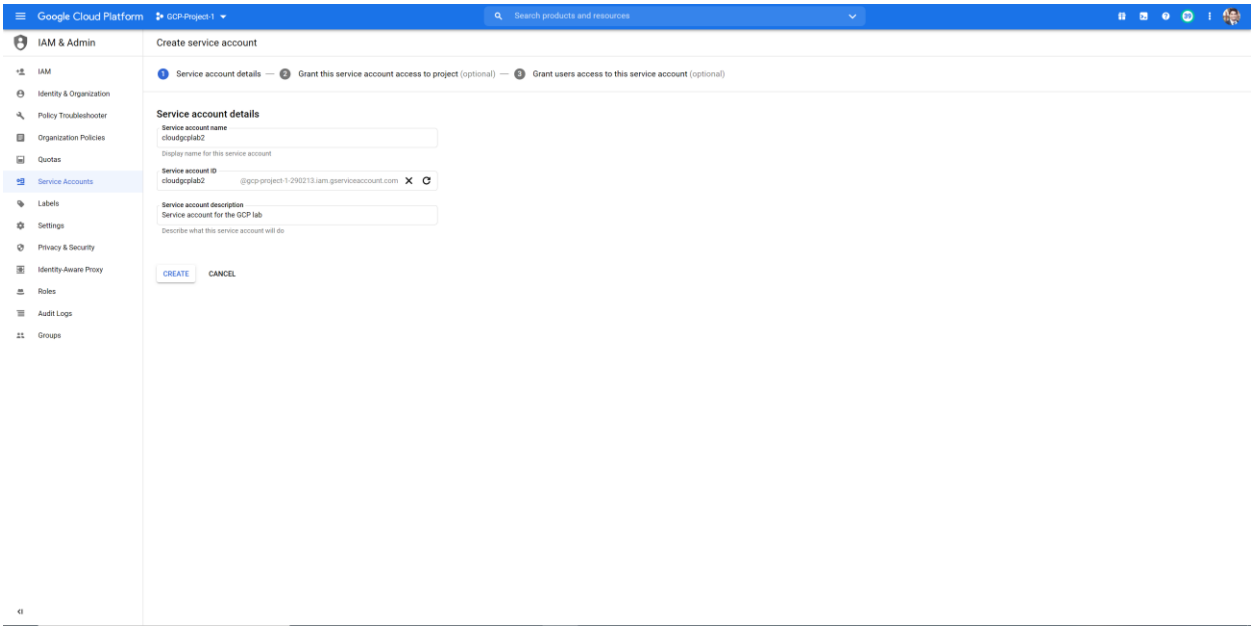


E. Setting Access Control

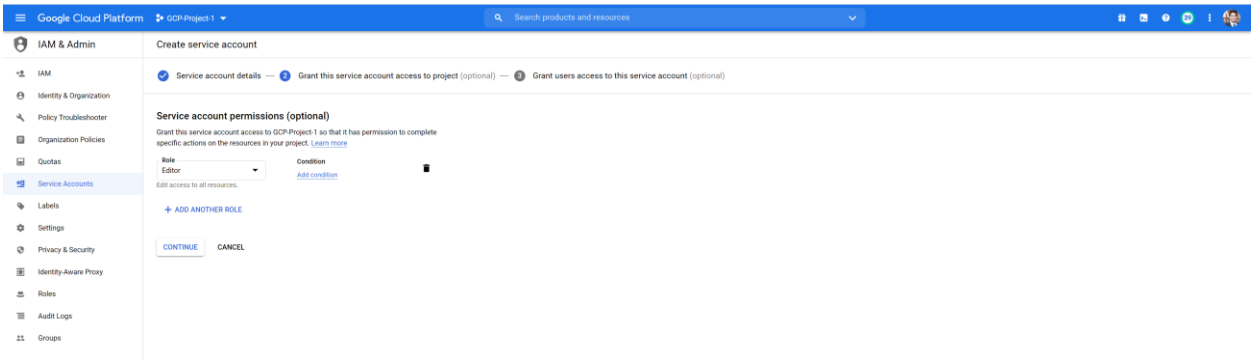
Complete the following sub-modules in ***Creating and enabling service accounts for instances:***

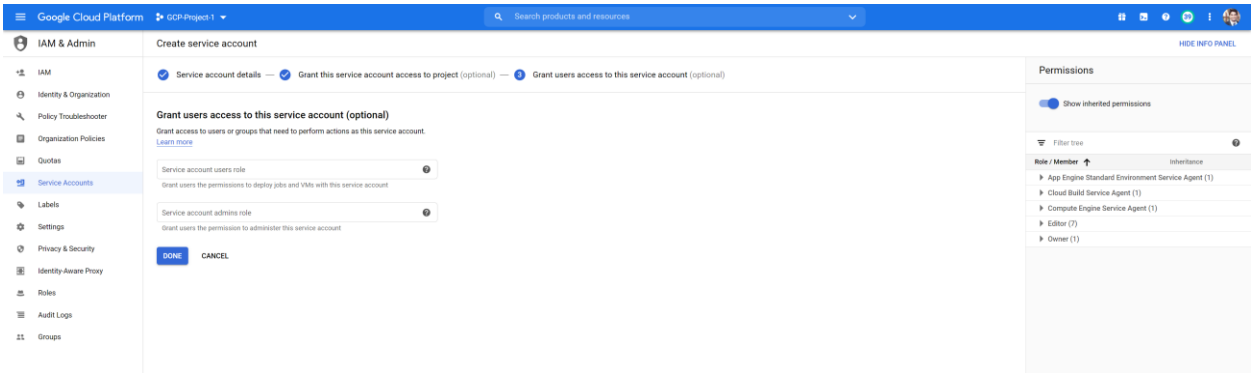
1. Creating a new service account

Service account is created to authenticate and run the applications using this service account instead of the user account. Service account is intended to represent a non-human user account which is authenticated and is authorized to access data. So, now we do not need to have every user authenticated to access the data by giving it privileges.

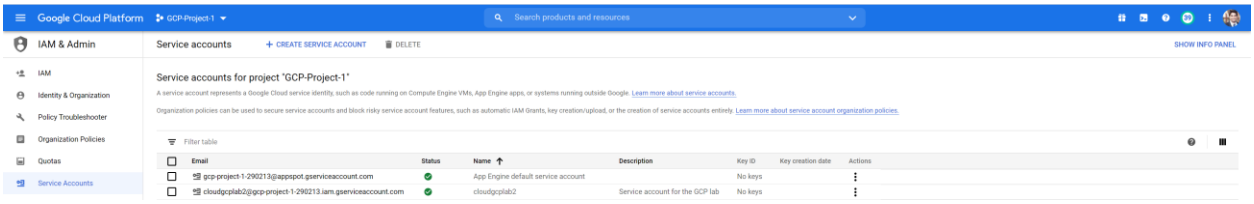


Service account is created with Editor role to be able to edit the resources of the VM.





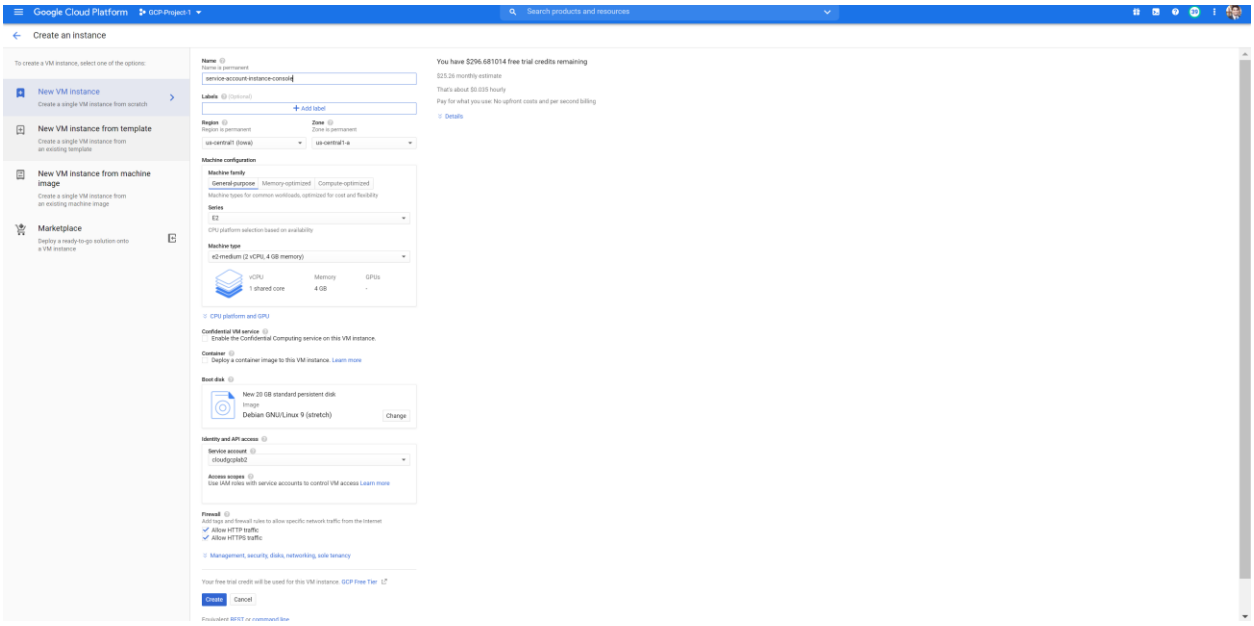
Service account is created successfully as shown below.

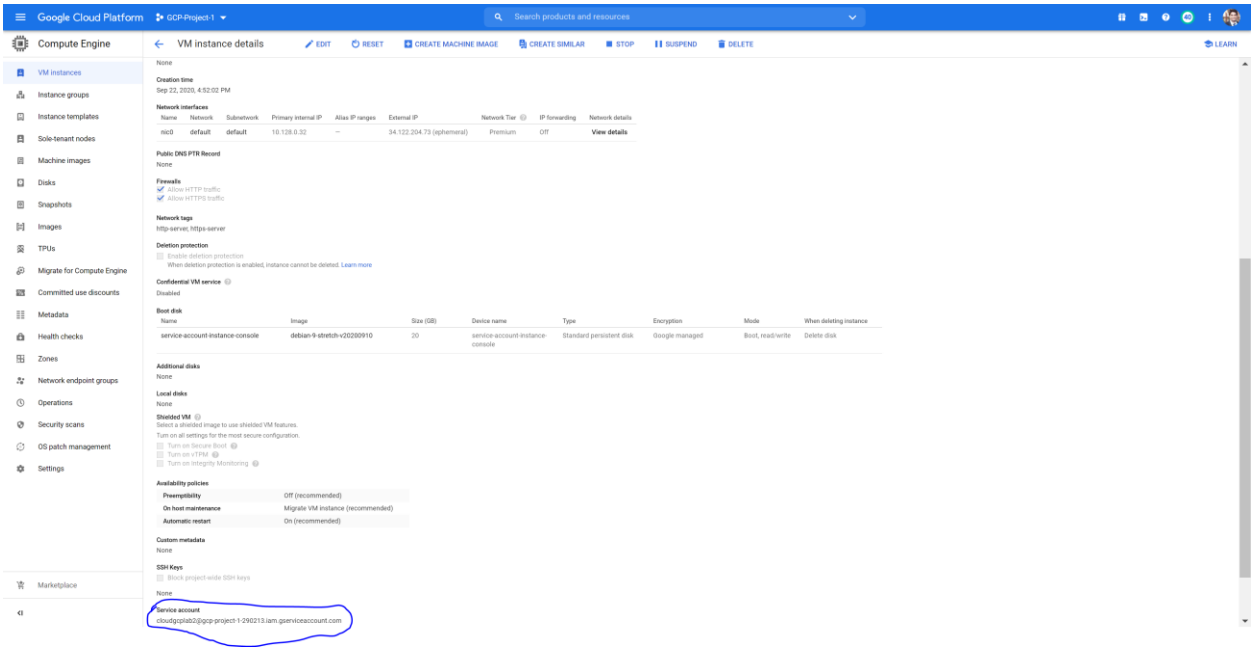


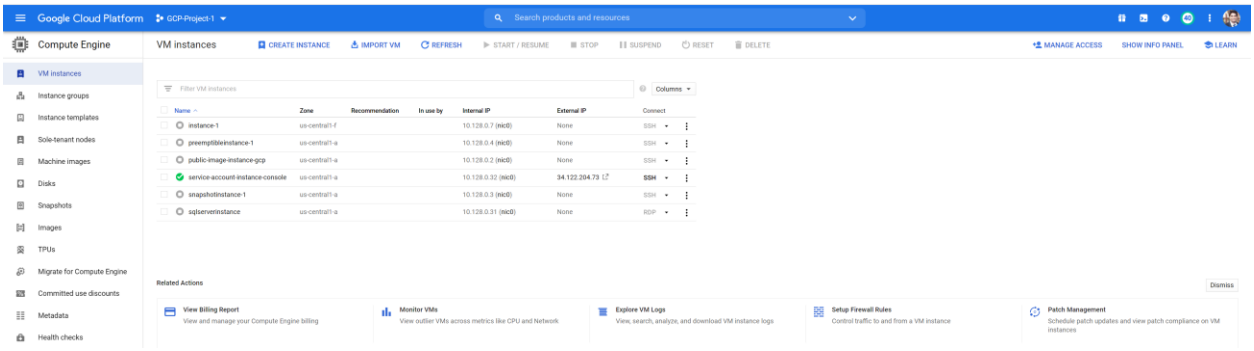
2. Setting up a new instance to run as a service account (Console and GCloud)

Console:

A new instance is created using the Console with the service account specified created earlier.

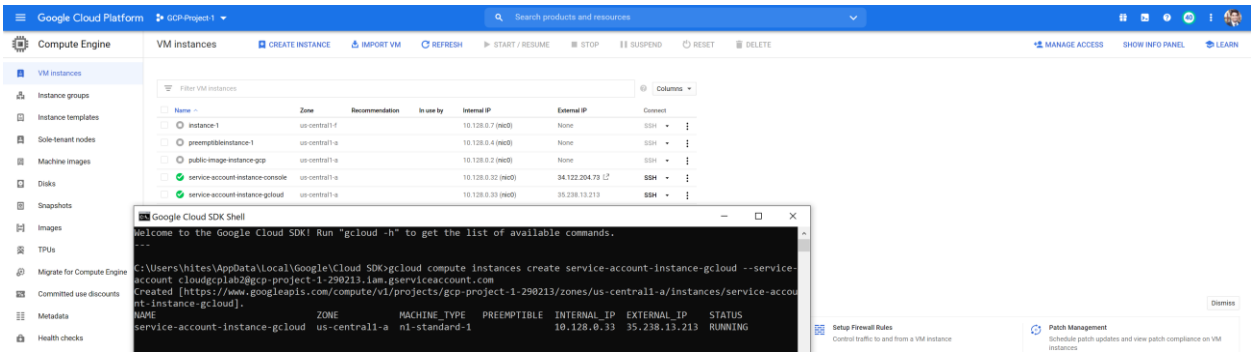






Gcloud:

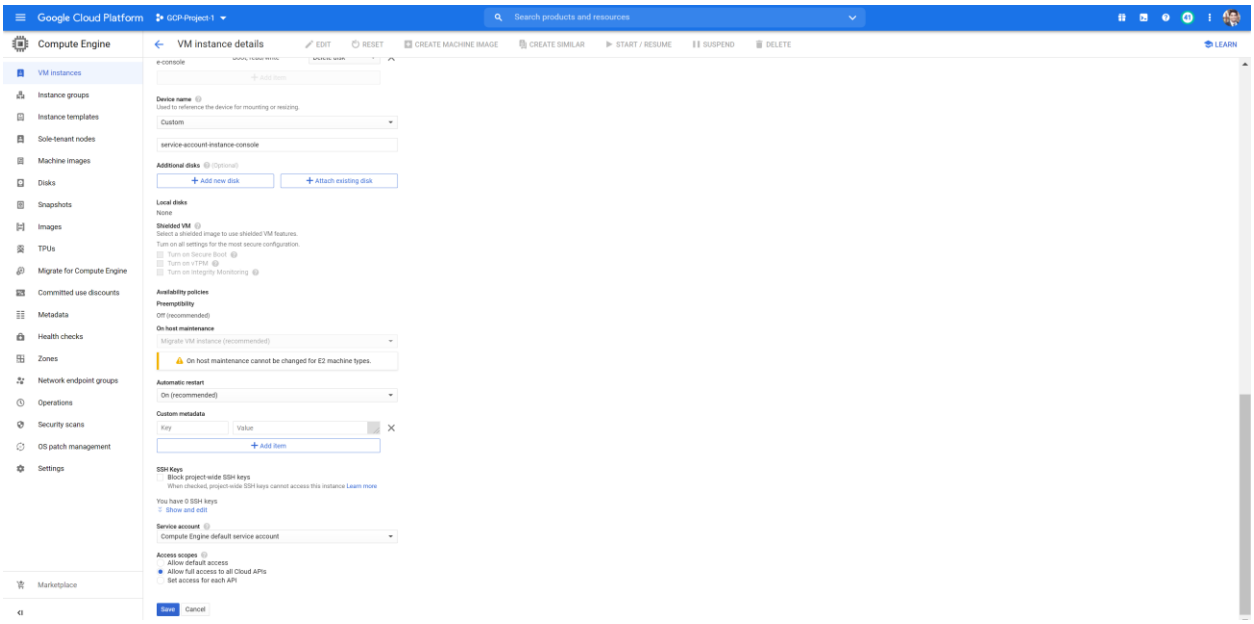
A new instance is created and assigned the service account using the Gcloud command line.

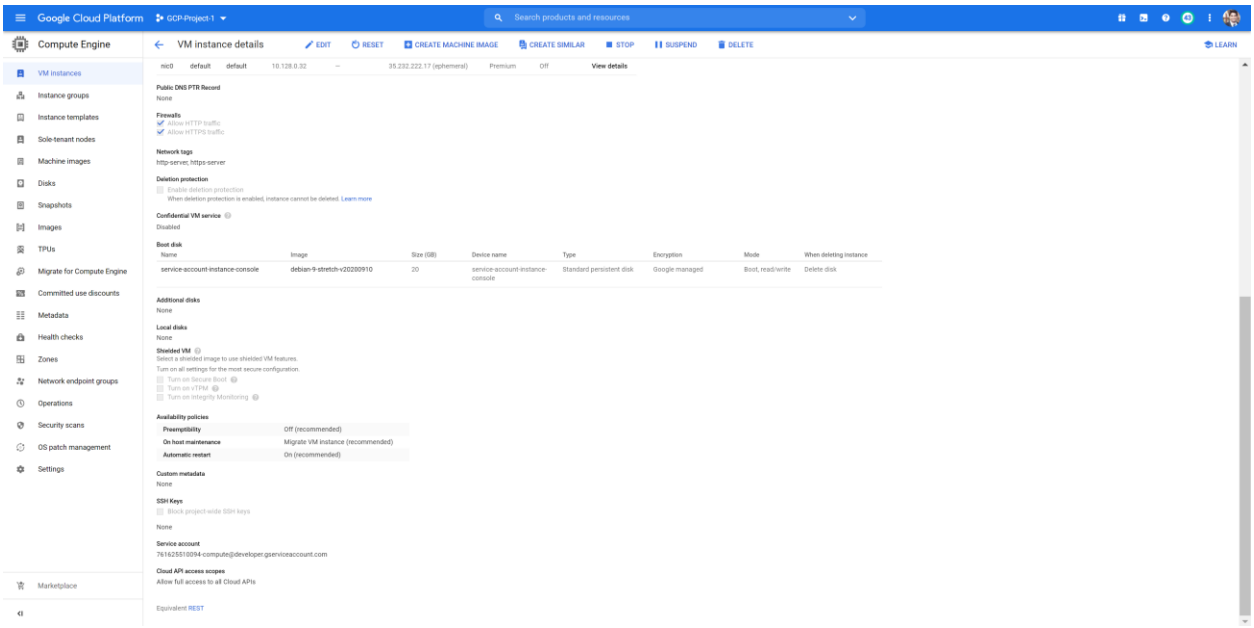


This service account with the Editor Roles allows to edit or modify the resources of the VM. All the instances with this service account linked to them would be able to edit the VM due to this Editor Role.

3. Changing the service account and access scopes for an instance

The service account for the instance is changed to the default Compute Engine service account and allowed full access to the Cloud API's so that there is a high degree of flexibility for the access which may be required to access the applications.





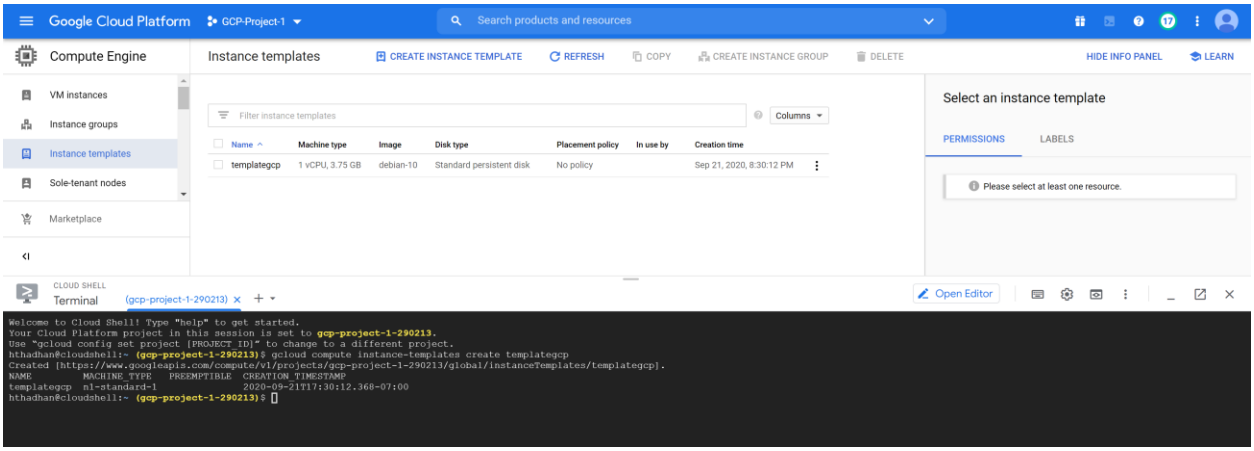
Section 6: Creating and Managing Instance Templates

Complete the following sub-modules in ***Creating Instance Templates***:

A. Creating a new instance template (GCloud)

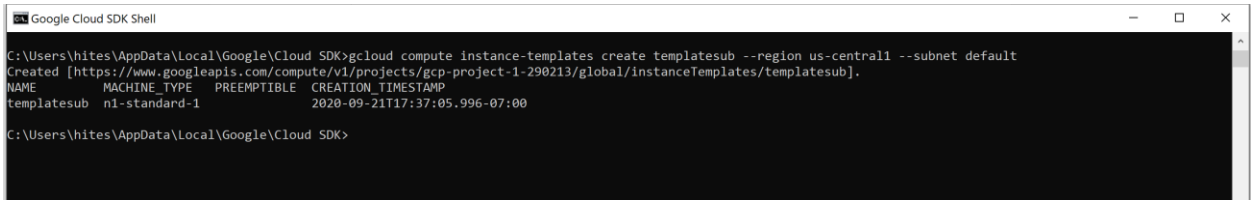
An instance template lets us define the machine type, boot disk image, network properties to create an instance later on using this template. These instance templates can be used to create a single individual instance or a group of managed instances. An instance template cannot be edited once created so for a modification we need to create a new template. These templates ease the creation process as we can define the parameters in the template to be used later on for the instance creation.

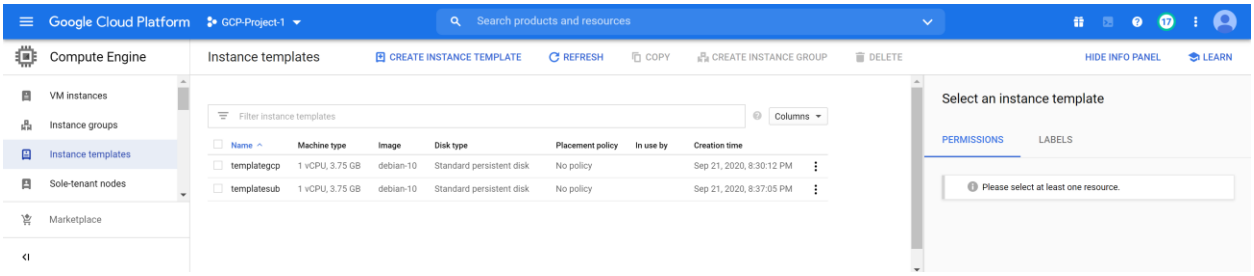
Instance template created using the Gcloud command line is shown below.



B. Creating an instance template that specifies a subnet

Now creating an instance template using a subnet. This allows to have an instance within a specified subnet or sub network which is defined earlier in the subnet and not on a global network.





Section 7: Creating and Managing Groups of Instances

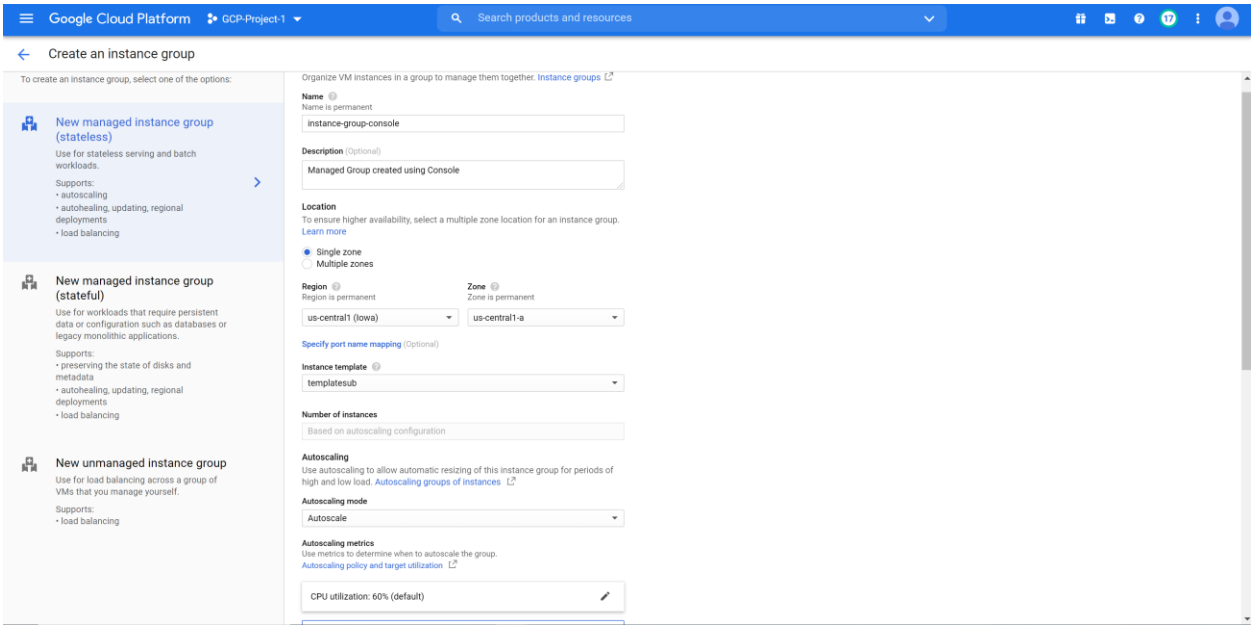
A. Complete the following sub-modules in ***Creating managed instance groups***:

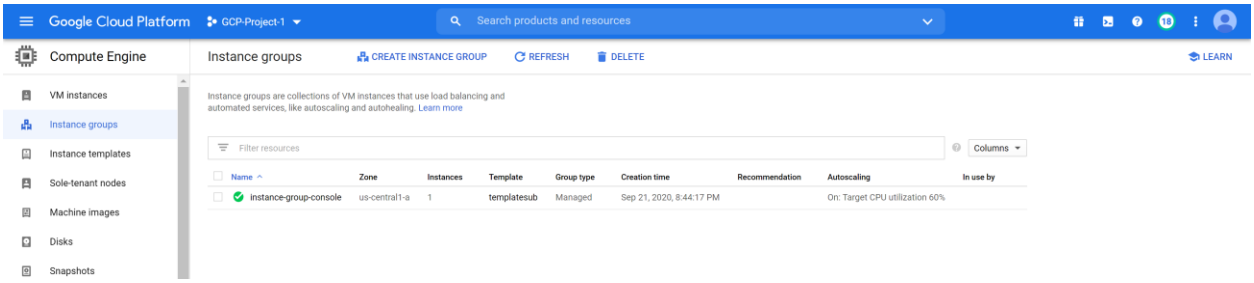
1. Creating a managed instance group (Console and GCloud)

Managed instance group of instances are a group of managed instances which are managed as a single entity and not as an individual instance. Any change would be applied to all the instances of the managed group.

Console:

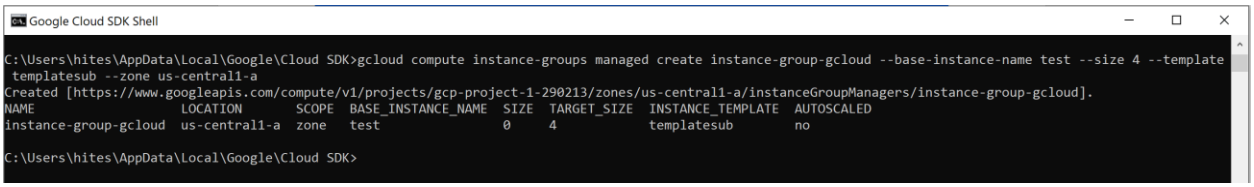
Created a Managed group of instances with the template templatesub created earlier using the Console.

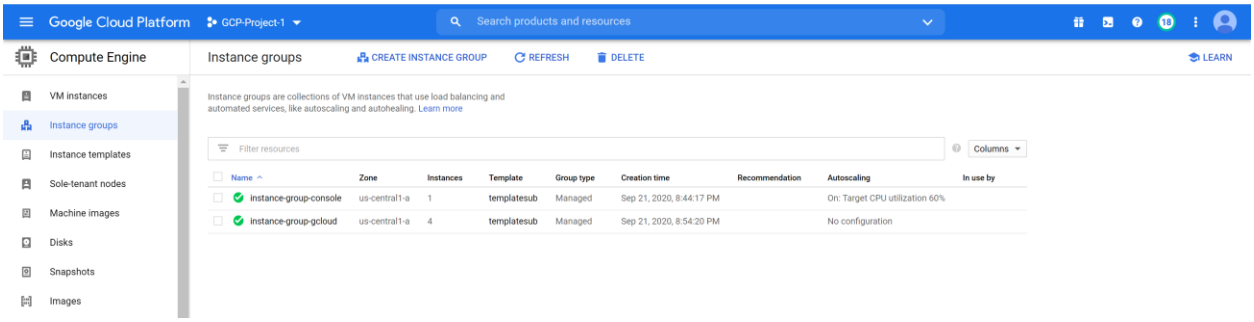




Gcloud:

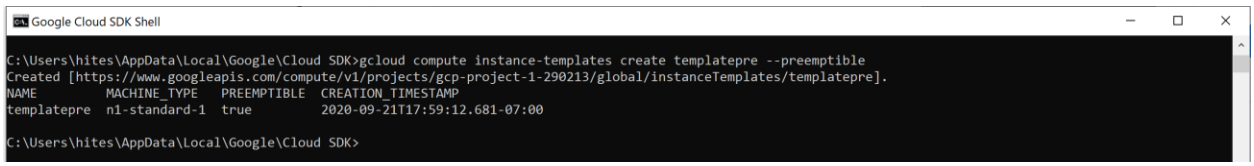
Managed group of instances is created using the Gcloud command line with managed as the word after instance-groups indicating the instances are managed group along with templatesub as the template to create the Managed Group of instances.

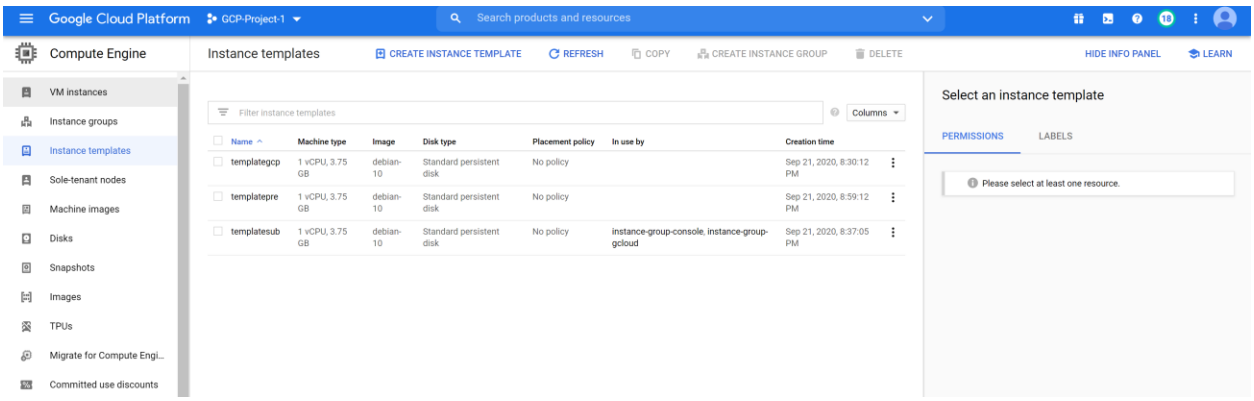




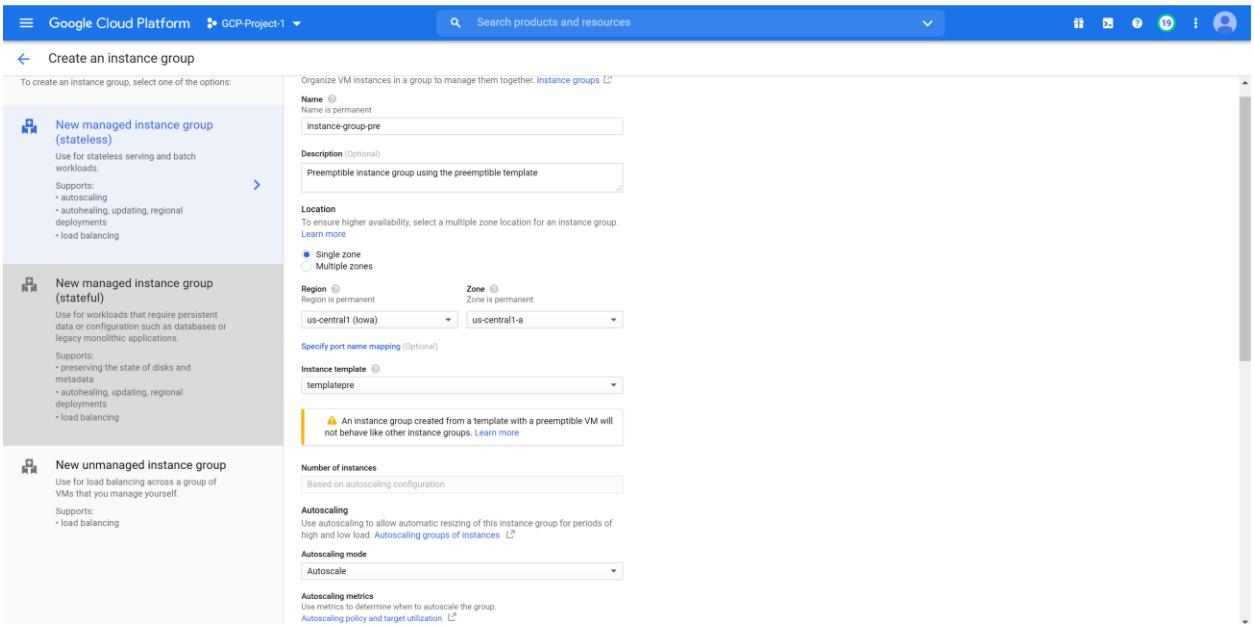
2. Creating groups of preemptible instances

Preemptible instance template is created first for it to be used as a template to create a Preemptible Group of managed instances using the Gcloud command line.

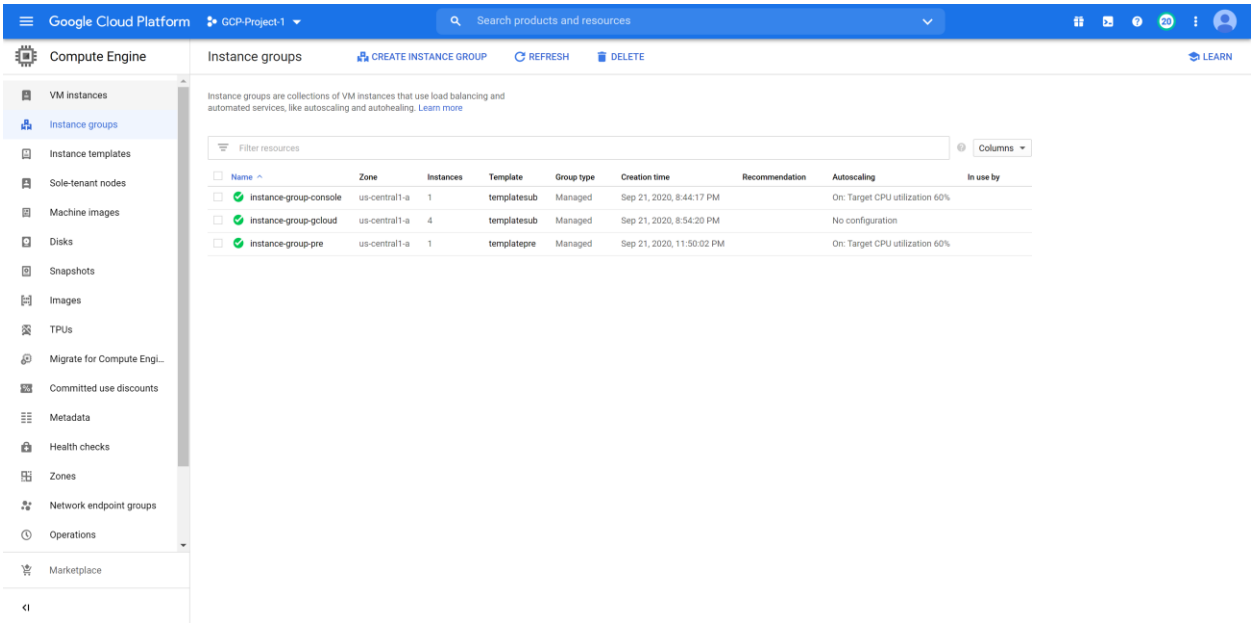


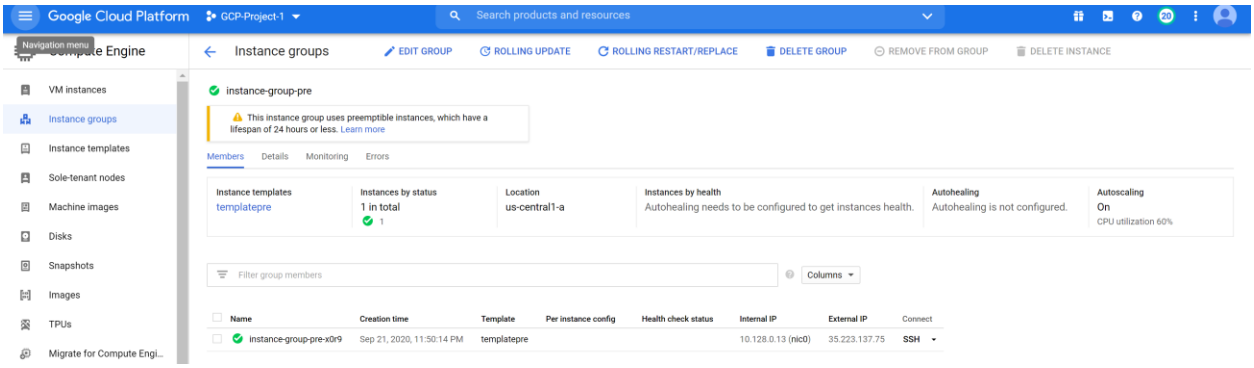


A new Managed Instance group is created with the template templatepre as the preemptible instance template.



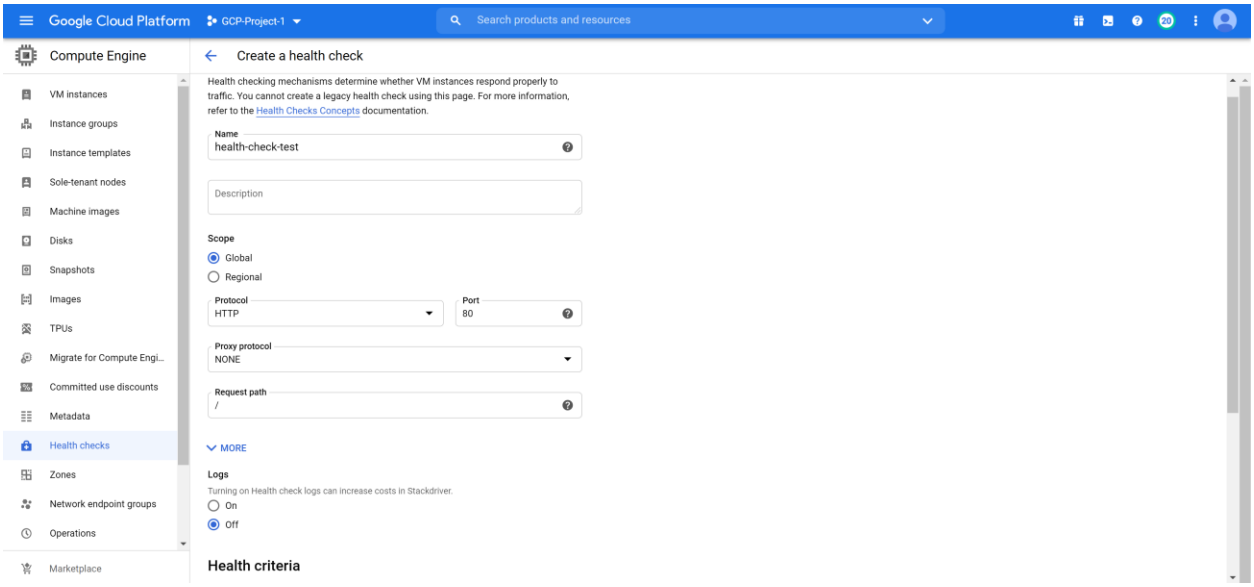
Preemptible instance group creation is successful as shown below.



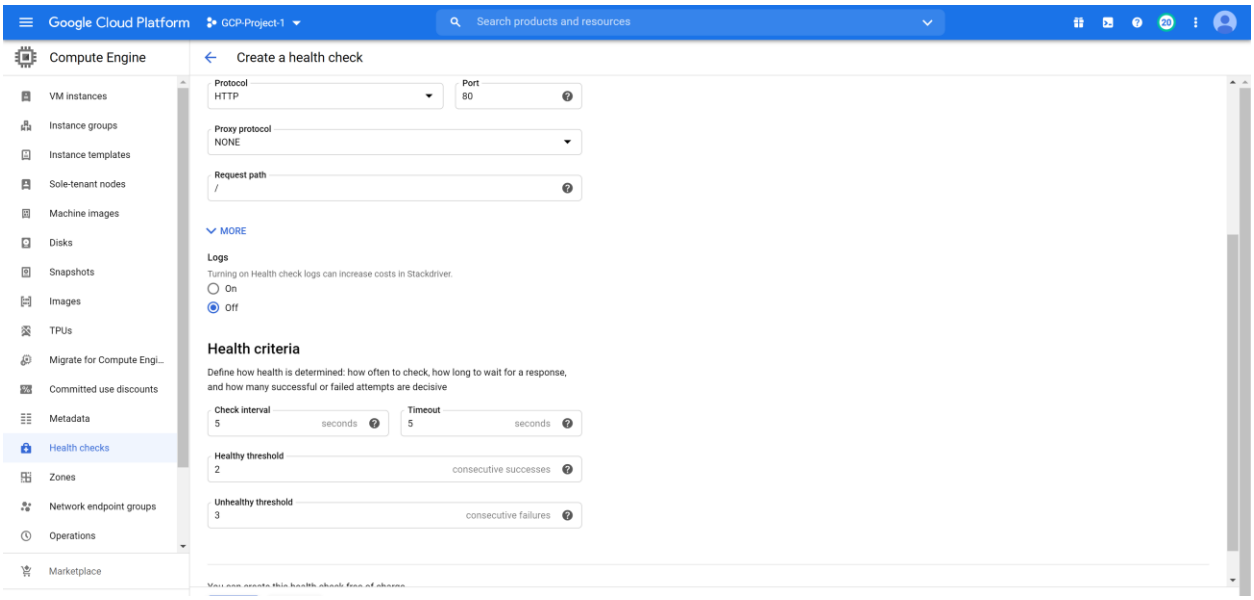


B. Complete the following sub-module in ***Setting up health checking and autohealing:*** Setting up a health check and an autohealing policy (Console and GCloud)

Here we are setting up a health check policy to look for any issues with the instance health, responding within a specified amount of time to detect any failure and automatically heal this failure. This health check rule is configured on the port 80 for checking the health of the instance with health check setting set below.



Health check setting is set with 5 sec intervals to check for the health. If there are 2 consecutive success responses on the port 80 from the instance it is considered to be healthy. If there are 3 consecutive failures of responses, the instance is considered UNHEALTHY and needs to be fixed to make it healthy again using the Console.

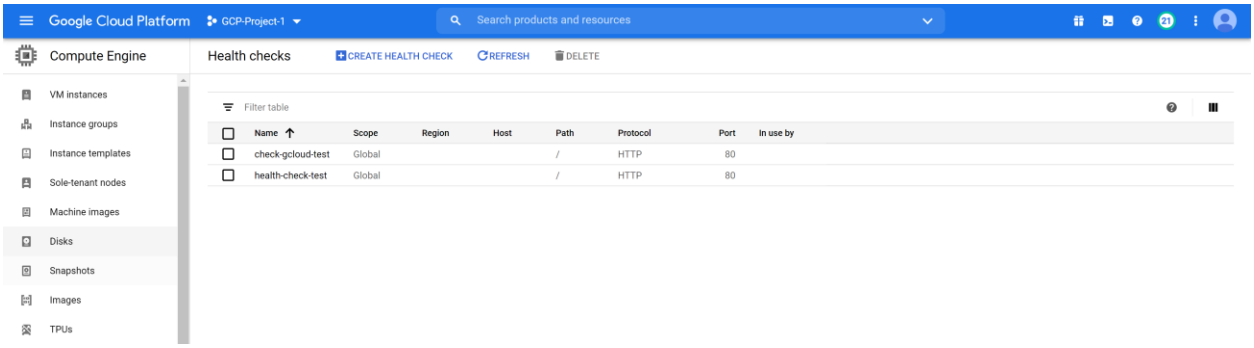


Setting up the health check using Gcloud command line with health check interval set to be 30sec to check the health every 30sec on the port 80 specified. Healthy threshold is set to 1 and 3 consecutive failures result in unhealthy state of the instance.

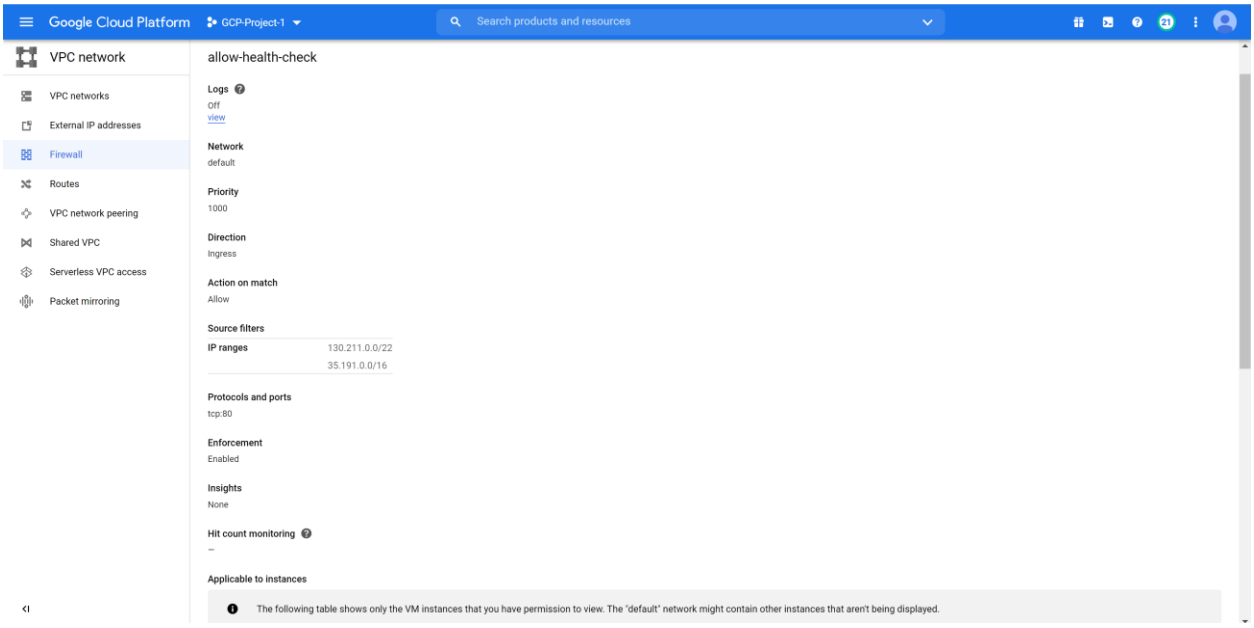
```
C:\Users\hites\AppData\Local\Google\Cloud SDK>gcloud compute health-checks create http check-gcloud-test --port 80 --check-interval 30s --healthy-threshold 1 --timeout
 10s --unhealthy-threshold 3
Created [https://www.googleapis.com/compute/v1/projects/gcp-project-1-290213/global/healthChecks/check-gcloud-test].
NAME              PROTOCOL
check-gcloud-test  HTTP

C:\Users\hites\AppData\Local\Google\Cloud SDK>
```

Firewall rule is created to allow the health-check-rule created earlier to connect to the application with the mentioned IP addresses where the IP Addresses are inspected for checking the health.

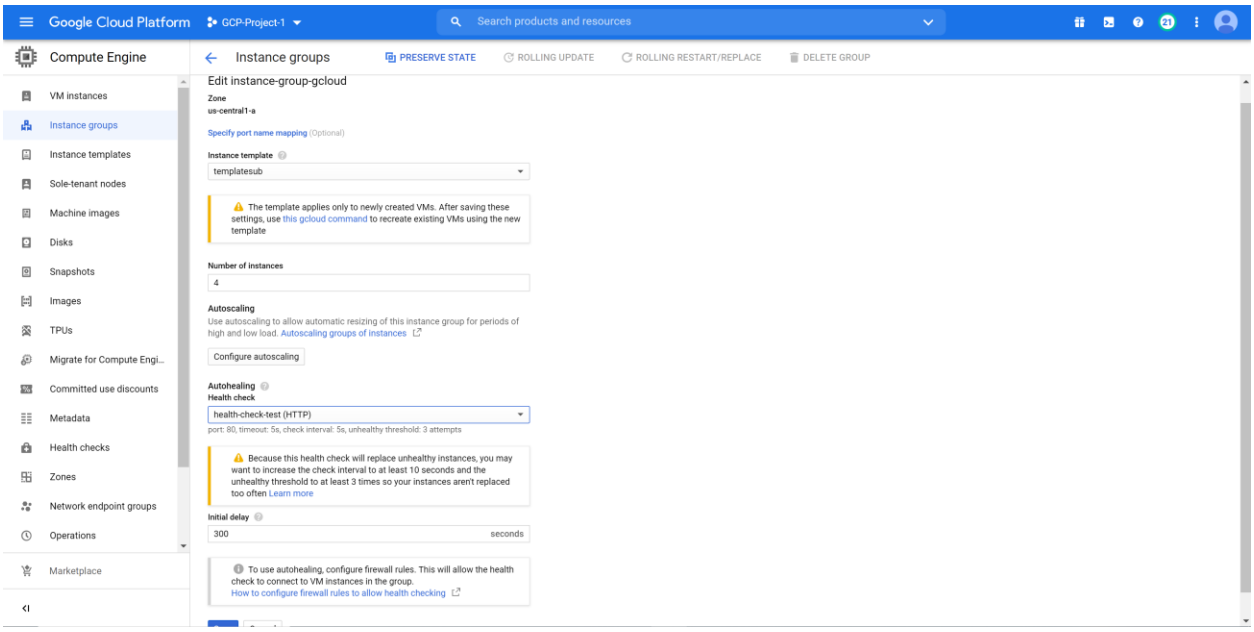
```
C:\Users\hites\AppData\Local\Google\Cloud SDK>gcloud compute firewall-rules create allow-health-check --allow tcp:80 --source-ranges 130.211.0.0/22,35.191.0.0/16
--network default
Creating firewall...-Created [https://www.googleapis.com/compute/v1/projects/gcp-project-1-290213/global/firewalls/allow-health-check].
Creating firewall...done.
NAME               NETWORK  DIRECTION  PRIORITY  ALLOW   DENY  DISABLED
allow-health-check  default  INGRESS    1000      tcp:80        False

C:\Users\hites\AppData\Local\Google\Cloud SDK>
```

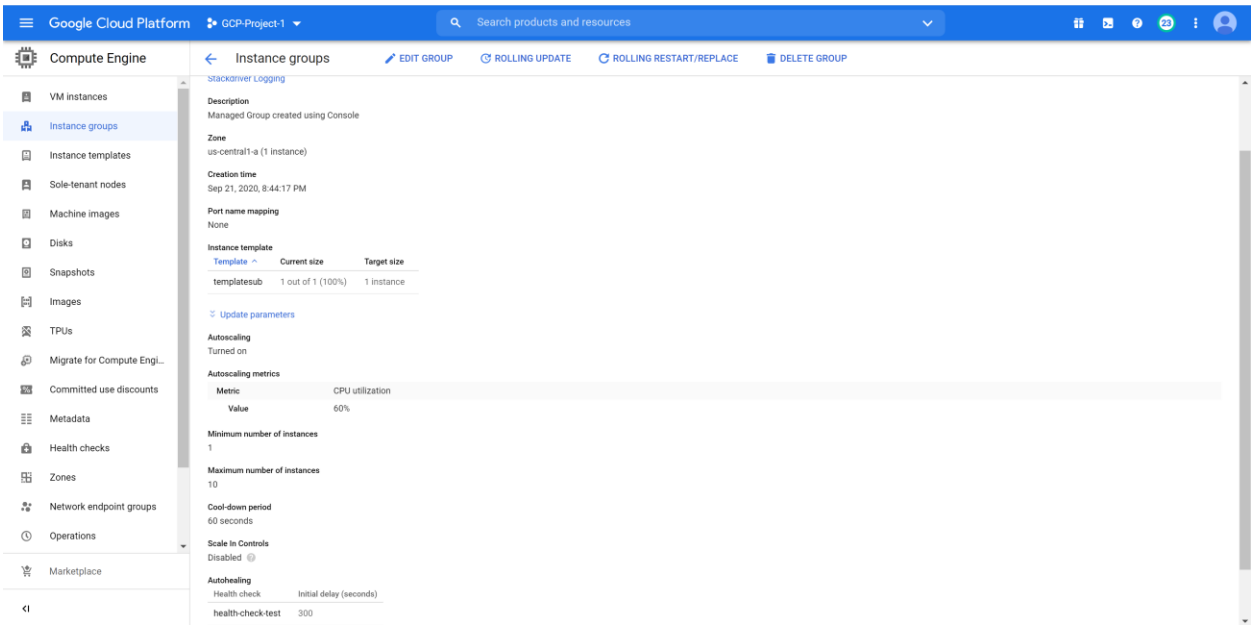


Health-check-rule created shows the IP Addresses to be checked for health of the instances in this range of the VM IP's.

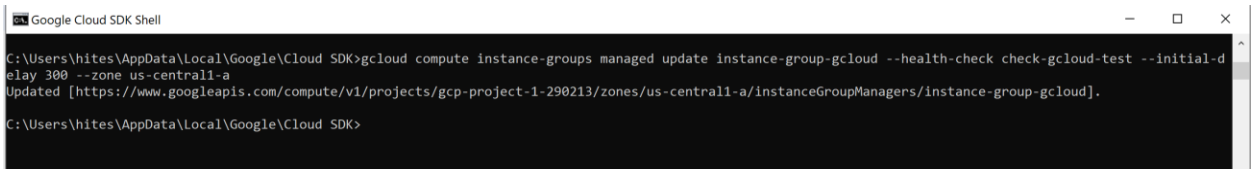


Assigning the Health checks created to the instances using the Console. In the Health Check, I mention the health check created earlier as the way of checking the health of the instance.

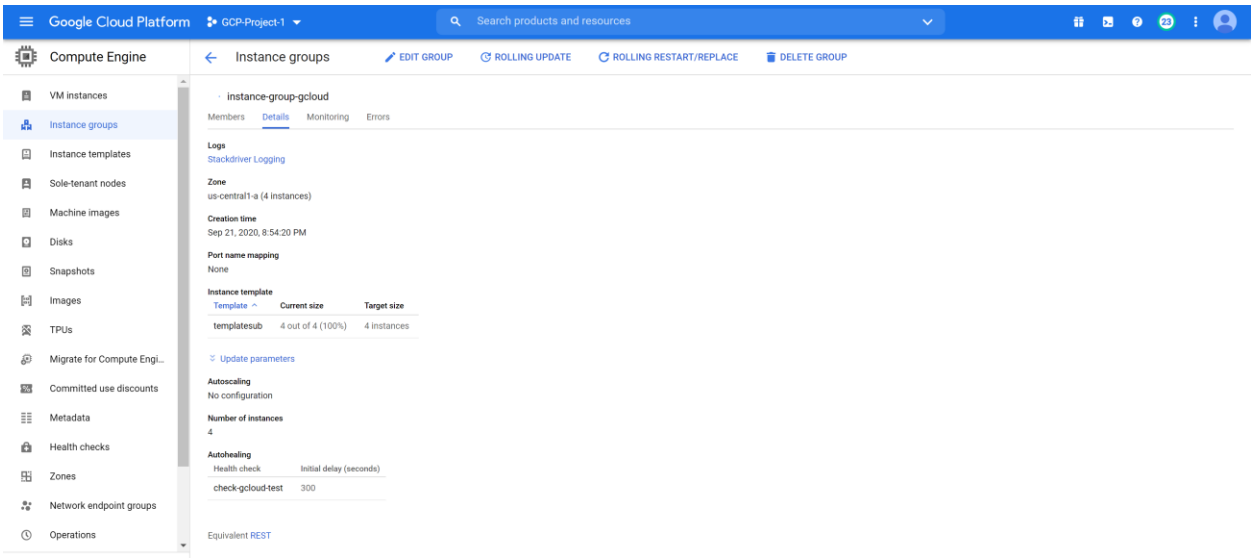




Updated the Managed Group created earlier with Health checks using Gcloud command line



The instance group is updated with the health check so that if the instance is UNHEALTHY to be able to find such Unhealthy instances.



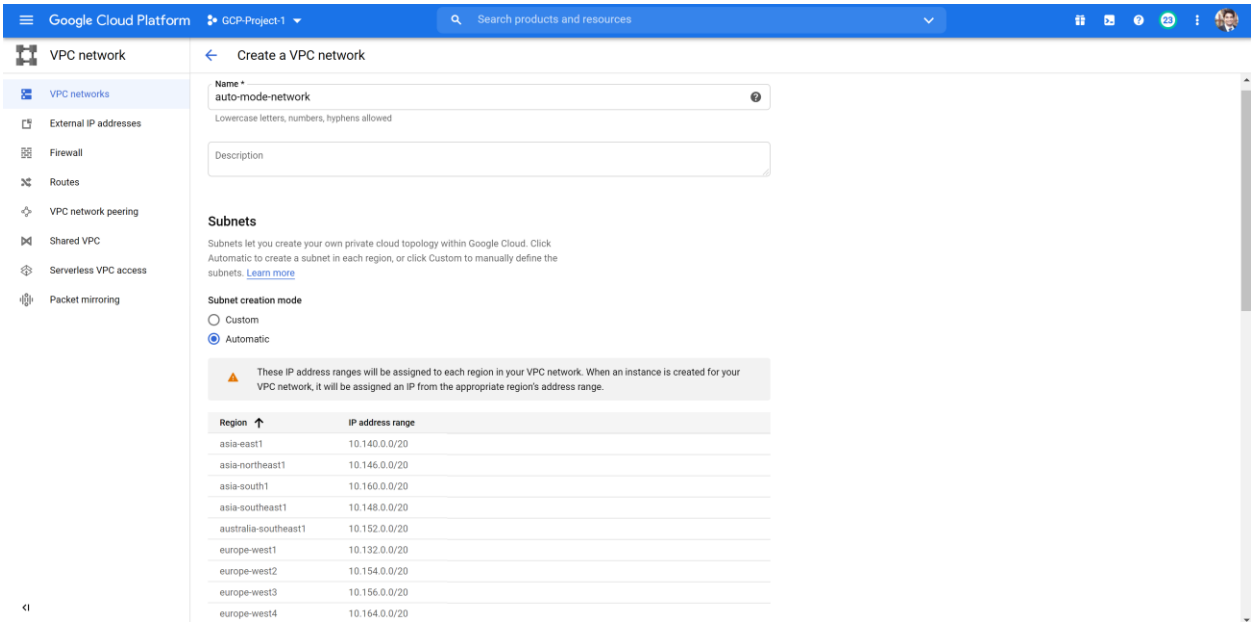
Section 8: Virtual Private Cloud (VPC)

A. Complete the following sub-modules in ***Using VPC network***s

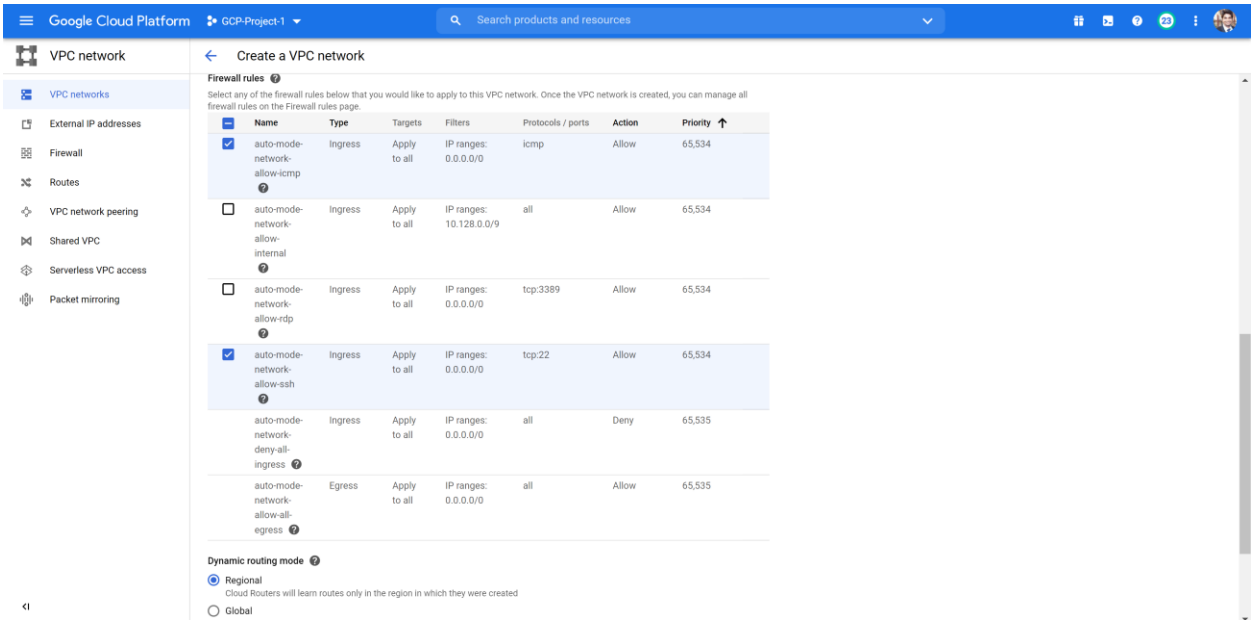
1. Creating networks

i. Creating an auto mode network (Console)

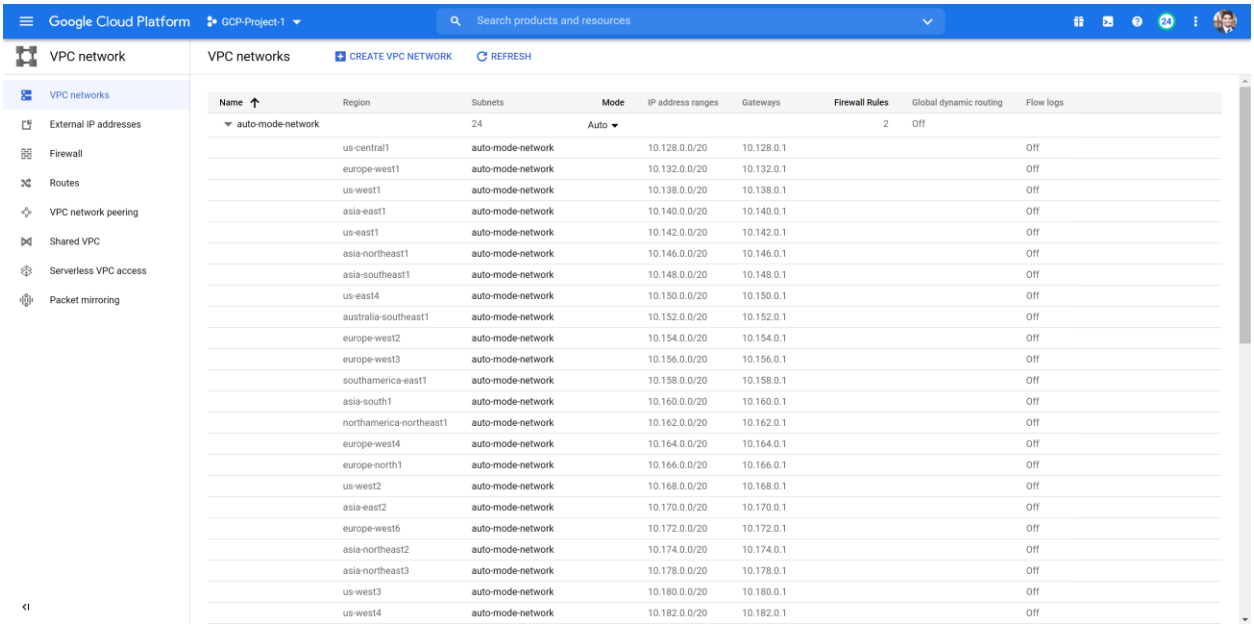
We are creating a VPC (Virtual Private Network) using the Auto mode Network. A VPC is a virtual version of the physical network implemented to create an isolated virtual network partition. In the Auto mode, all the regions are selected and 1 subnet each for each region is selected. As the new regions are added, subnet from these regions is added automatically.



Selected some firewall rules to allow routing to happen regionally.

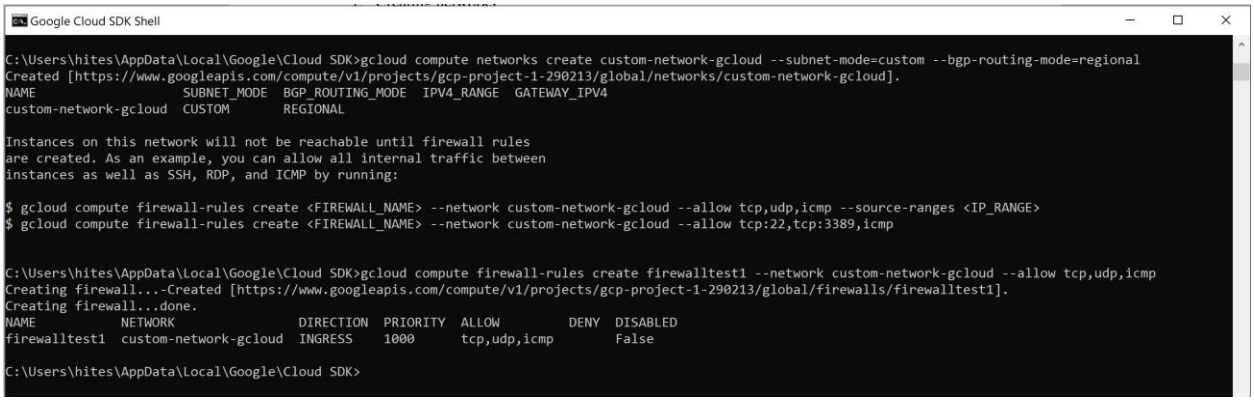


New auto mode network is created with 1 subnet from the each of the regions as shown below.

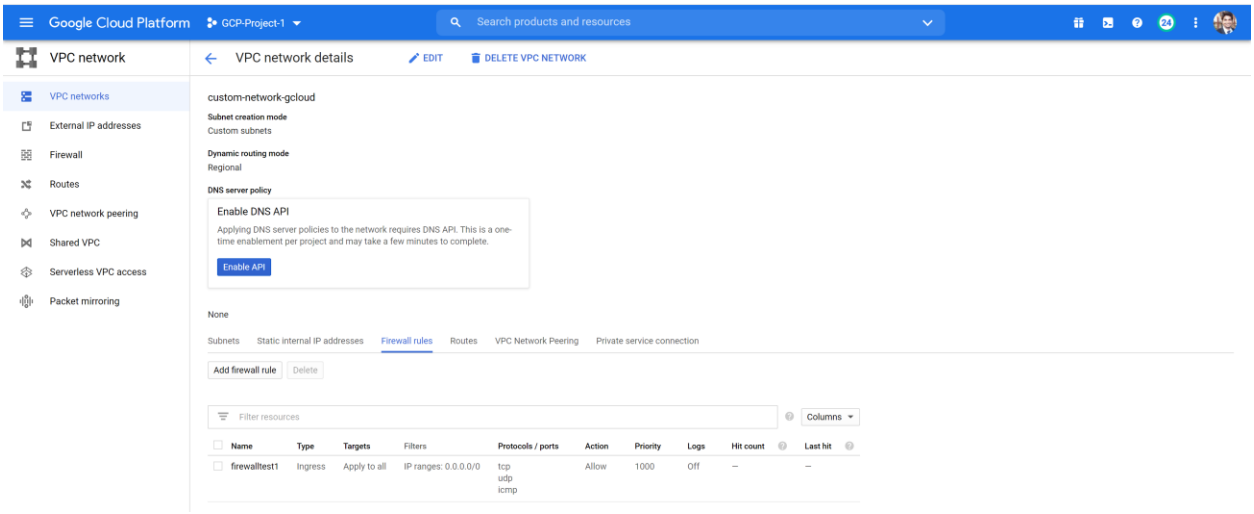


ii. Creating a custom mode network (GCloud)

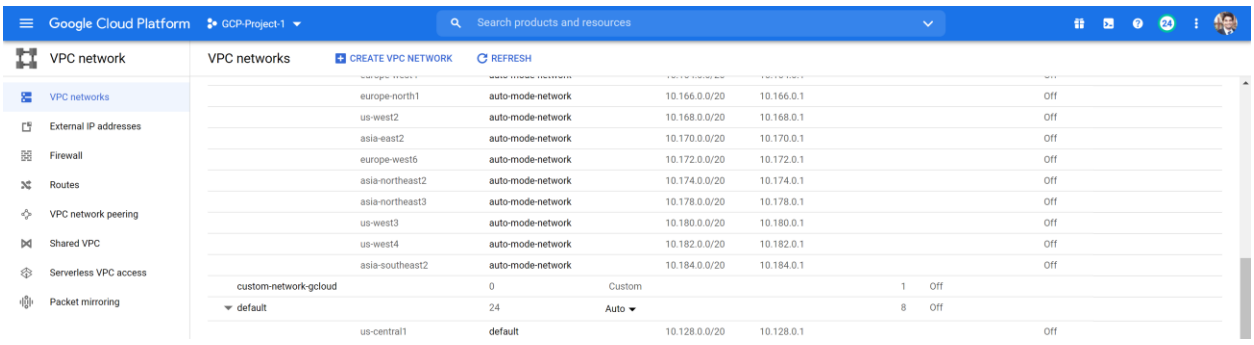
Created a custom network manually by setting subnet-mode as Custom along with Regional Routing enabled using the bgp-routing-mode=regional. This custom network created is set to allow the TCP, UDP and ICMP traffic after the creation.



Creation of the manual custom mode network with the required firewall rules is successful.



This shows the custom mode network with all the firewall rules and the region.

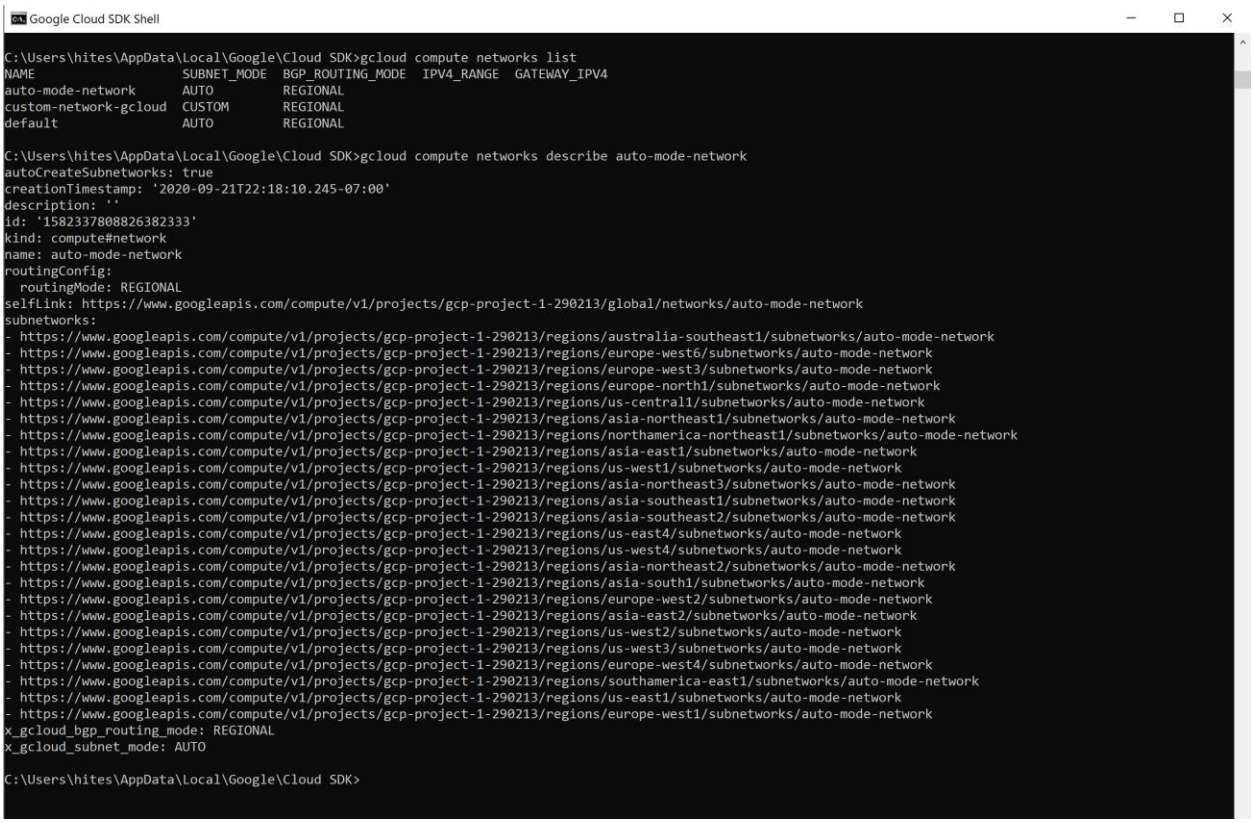


2. Viewing networks

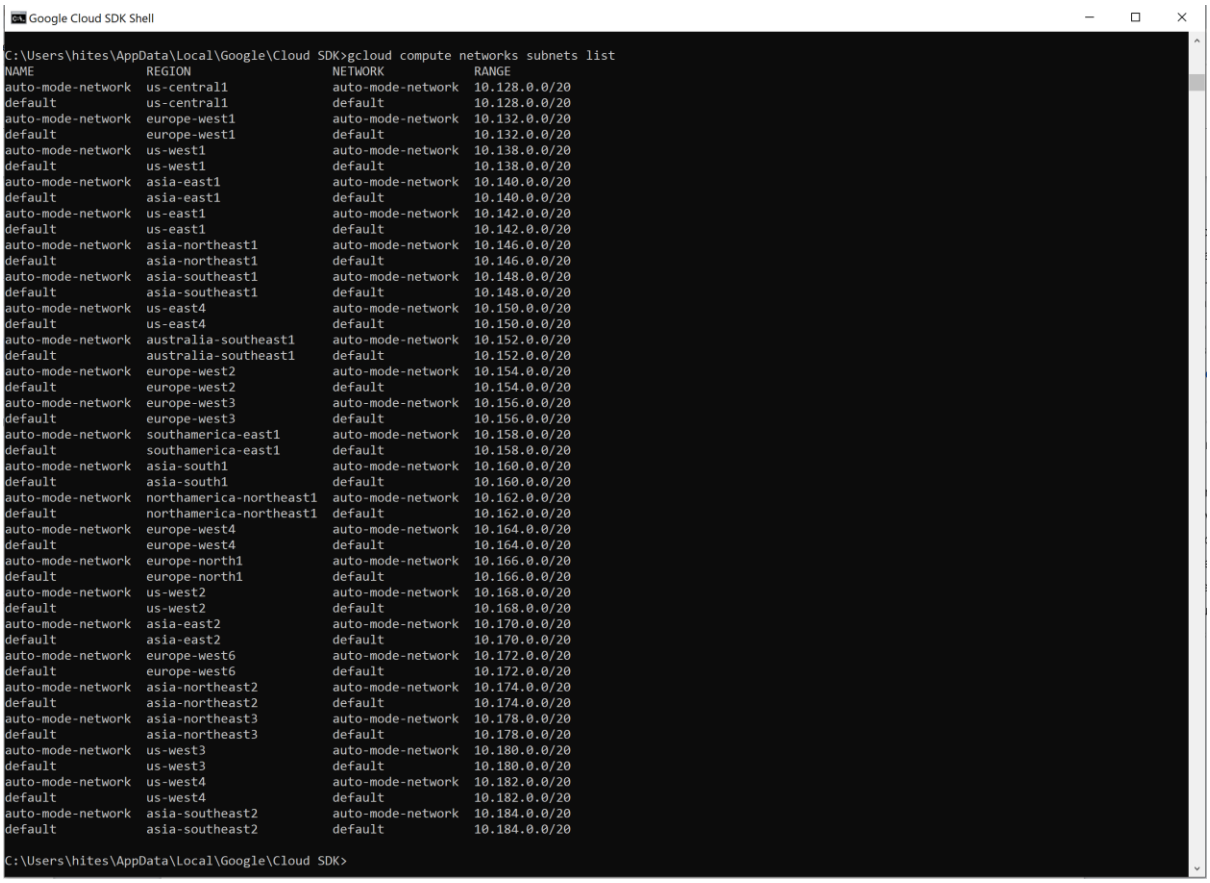
Viewing the VPC's and legacy network in the projects. For VPC's, we can view the information of the subnets and their subnet creation mode.

gcloud compute networks list – lists all the networks created. Here there are 3 networks with 1 default network and the other 2 networks being the auto mode and custom mode network created earlier.

gcloud compute networks describe auto-mode-network – describes a single network in detail showing the creation time, routing configuration and the subnetwork details for that network.

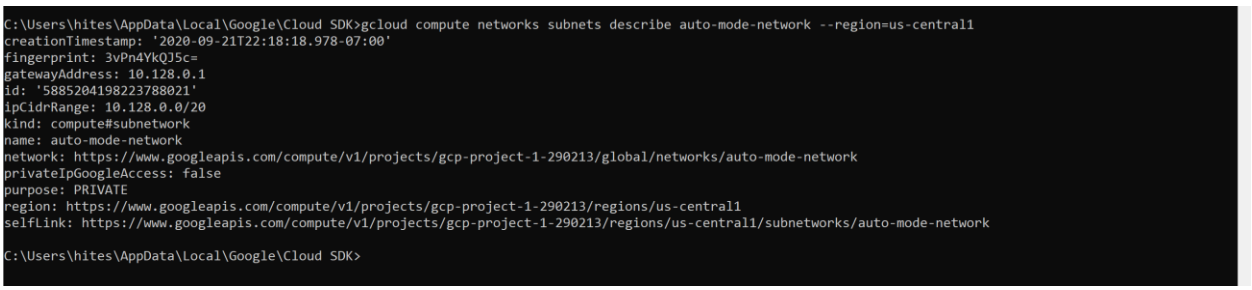


3. Working with subnets
    a. Listing subnets (GCloud)
    Listing all the subnets which are part of a project to know the range of IP Addresses which we can use for the instance in a particular subnet as each VPC is sub-divided into subnets. It also gives the details about the names and the regions of the existing subnets i.e. in which region which VPC has what range of IP addresses.



b. Describing a subnet (GCloud)

Describing a subnet gives detail information about the subnet such as primary IP Range, name of the network, region and any secondary IP ranges.

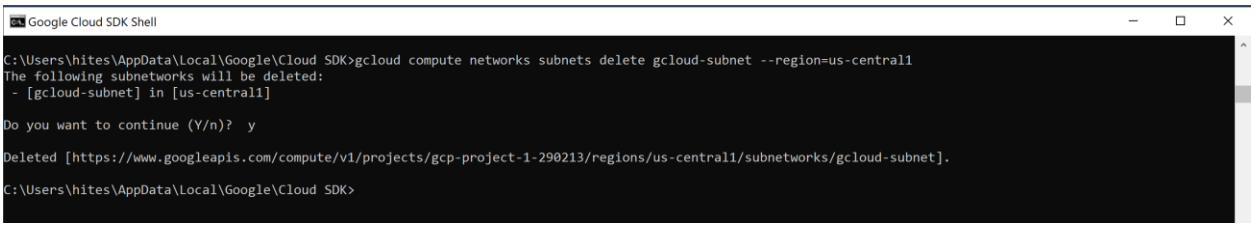


c. Adding subnets (GCloud)

We can create a new subnet in the given network with the name and Primary IP address range.

d. Deleting subnets (GCloud)

Deleting the subnets manually requires all the resources using these subnets to be deleted first such as delete VM's, internal firewall rules etc. Here there are no resources which are using this subnet so we can directly delete this subnet without requiring to delete any dependencies before deleting the subnet.

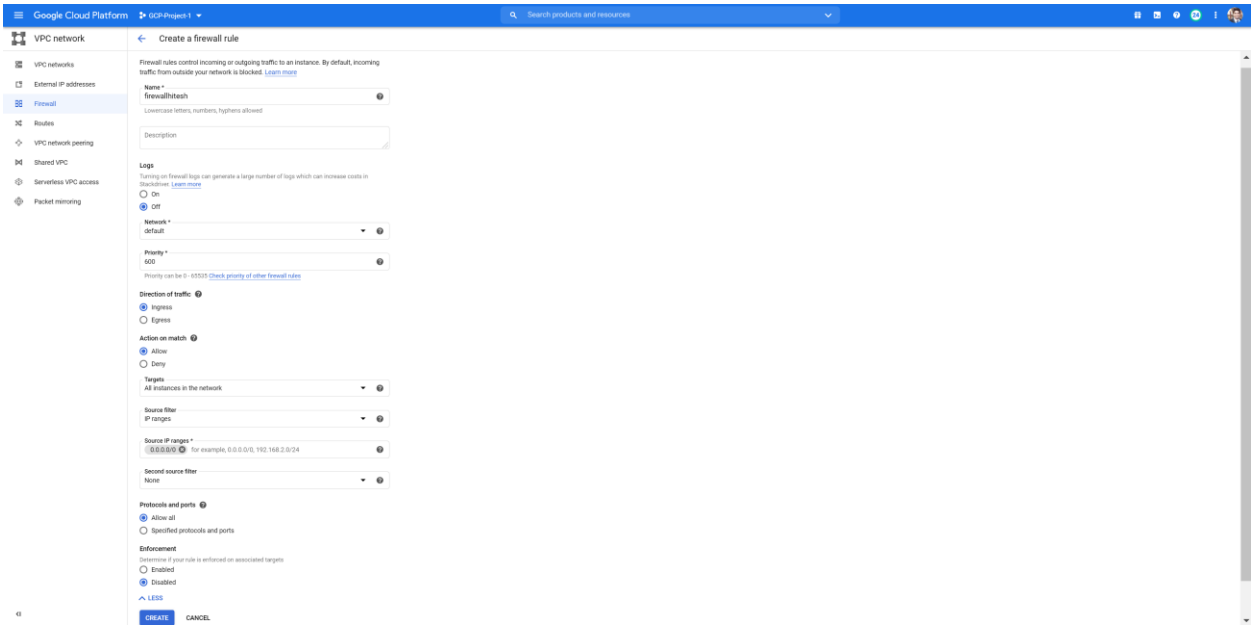


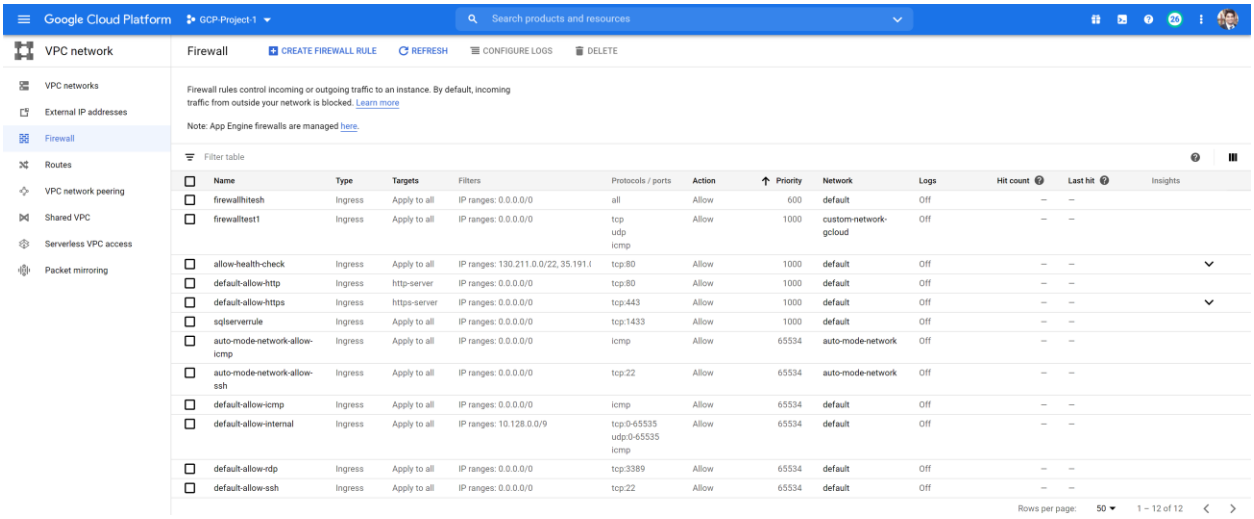
4. Modifying networks
    a. Deleting a network (GCloud)

If a network is not being used, we can delete it. If there are certain resources using this network, we would need to delete all the resources using this network first in order to be able to delete the network.

B. Complete the following sub-modules in ***Using firewall rules***:

1. Creating Firewall Rules (Console)
   An example of the firewall rule created with the configuration to allow Ingress traffic from the selected IP's.





2. Configuration examples - Recreate sample network configurations in the scenarios listed below

a. Example 1: Deny all ingress TCP connections except those to port 80 from subnet1

Firewall rule denies all the ingress traffic coming to port 80 from the subnet1 with a priority of 1000 using the Gcloud Command line.

```
C:\Users\hites\AppData\Local\Google\Cloud SDK>gcloud compute firewall-rules create example1-deny-access-subnet1 --network auto-mode-network --action deny
--direction ingress --rules tcp --source-ranges 0.0.0.0/0 --priority 1000 --target-tags webserver
Creating firewall...-Created [https://www.googleapis.com/compute/v1/projects/gcp-project-1-290213/global/firewalls/example1-deny-access-subnet1].
Creating firewall...done.
NAME                        NETWORK           DIRECTION  PRIORITY  ALLOW  DENY  DISABLED
example1-deny-access-subnet1  auto-mode-network  INGRESS    1000            tcp   False
```

Firewall to accept the TCP Ingres traffic on the port 80 from subnet1with priority 50 using Gcloud Command line. This implementation will allow TCP Ingres traffic to port 80 from subnet1 as a result of the priorities set for the rules created. This links these firewall rules created to the specified network auto-mode-network.

```
C:\Users\hites\AppData\Local\Google\Cloud SDK>gcloud compute firewall-rules create example1-allow-access-port80-from-subnet1-ingress --network auto-mode-n
etwork --action allow --direction ingress --rules tcp:80 --source-ranges 10.240.10.0/24 --priority 50 --target-tags webserver
Creating firewall...-Created [https://www.googleapis.com/compute/v1/projects/gcp-project-1-290213/global/firewalls/example1-allow-access-port80-from-subne
t1-ingress].
Creating firewall...done.
NAME                                            NETWORK           DIRECTION  PRIORITY  ALLOW   DENY  DISABLED
example1-allow-access-port80-from-subnet1-ingress  auto-mode-network  INGRESS    50        tcp:80        False

C:\Users\hites\AppData\Local\Google\Cloud SDK>
```

b. Example 2: Deny all egress TCP connections except those to port 80 of vm1

Firewall rule denies all the egress traffic coming to port 80 from the vm1 with a priority of 1000 using the Gcloud Command line.

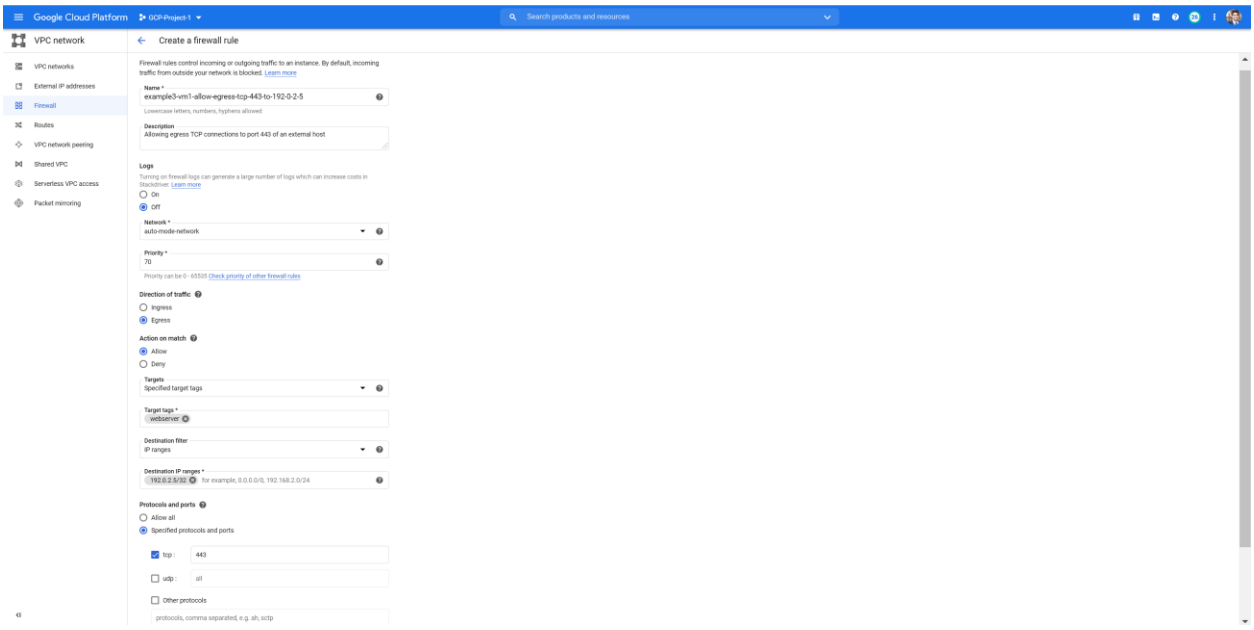
```
C:\Users\hites\AppData\Local\Google\Cloud SDK>gcloud compute firewall-rules create example2-deny-all-access-egress --network auto-mode-network --action de
ny --direction egress --rules tcp --destination-ranges 0.0.0.0/0 --priority 1000
Creating firewall...-Created [https://www.googleapis.com/compute/v1/projects/gcp-project-1-290213/global/firewalls/example2-deny-all-access-egress].
Creating firewall...done.
NAME                           NETWORK           DIRECTION  PRIORITY  ALLOW  DENY  DISABLED
example2-deny-all-access-egress  auto-mode-network  EGRESS     1000            tcp   False
```

Firewall to accept the TCP Egress traffic on the port 80 from vm1 with priority 60 using Gcloud Command line. This implementation will allow TCP egress traffic to port 80 from vm1 as a result of the priorities set for the rules created. This links these firewall rules created to the specified network auto-mode-network.

```
C:\Users\hites\AppData\Local\Google\Cloud SDK>gcloud compute firewall-rules create example2-vm1-allow-egress-port80-tcp --network auto-mode-network --acti
on allow --direction egress --rules tcp:80 --destination-ranges 192.168.1.2/32 --priority 60
Creating firewall...-Created [https://www.googleapis.com/compute/v1/projects/gcp-project-1-290213/global/firewalls/example2-vm1-allow-egress-port80-tcp].
Creating firewall...done.
NAME                              NETWORK           DIRECTION  PRIORITY  ALLOW   DENY  DISABLED
example2-vm1-allow-egress-port80-tcp  auto-mode-network  EGRESS     60        tcp:80        False
```

c. Example 3: Allow egress TCP connections to port 443 of an external host

Firewall rule created to allow all the TCP egress traffic to port 443 from an external host using the Console. This links this firewall rule to the network auto-mode-network with the priority as 70 as specified in the below Console configurations.



d. Example 4: Allow SSH connections from vm2 to vm1

Firewall rule created to allow SSH connection traffic from vm2 to vm1 on the default port of SSH which is 22 for the auto-mode-network using the gcloud command line.