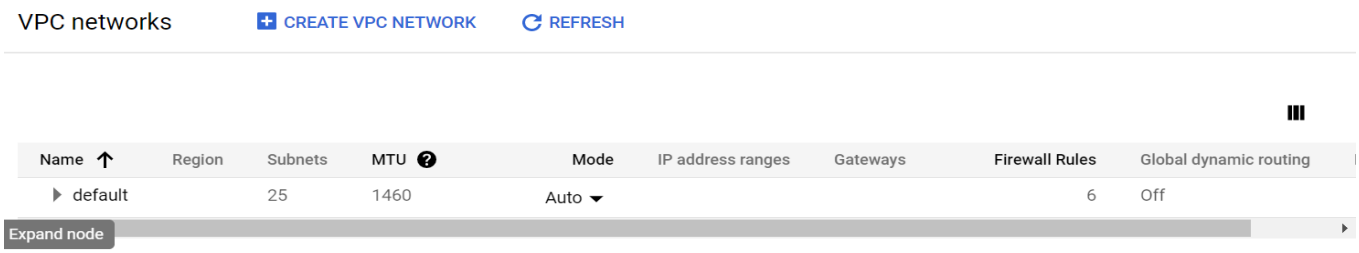
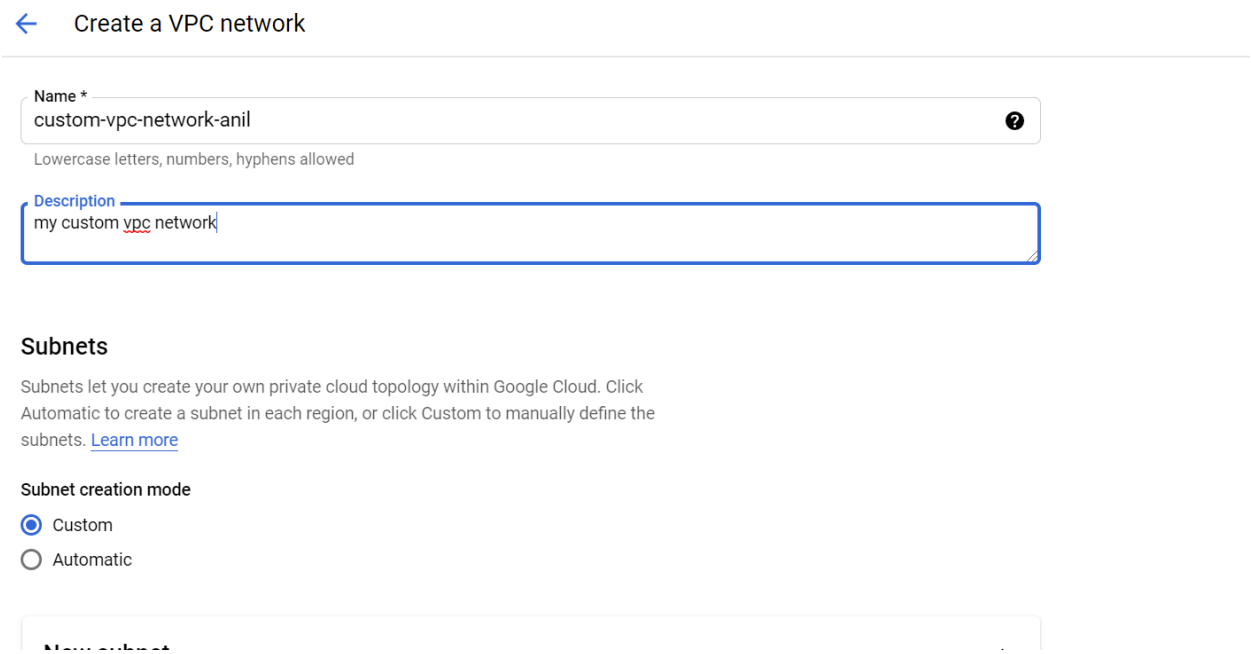
1) **Create a VPC network in Custom mode**

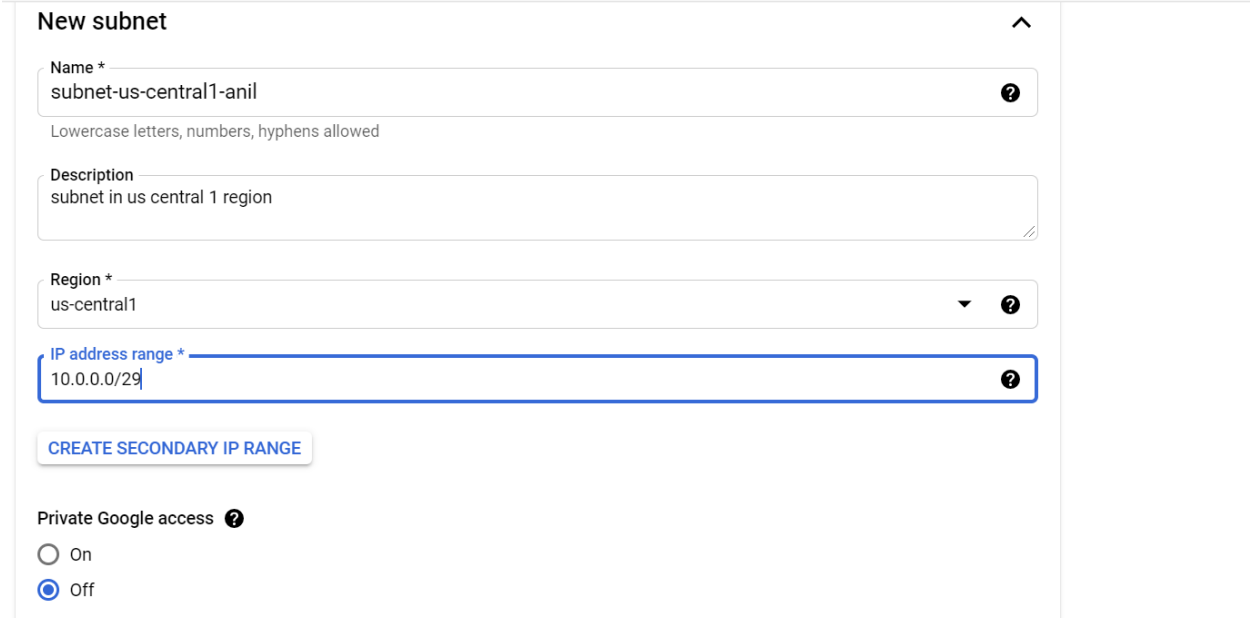
Go to VPC network and click create VPC network



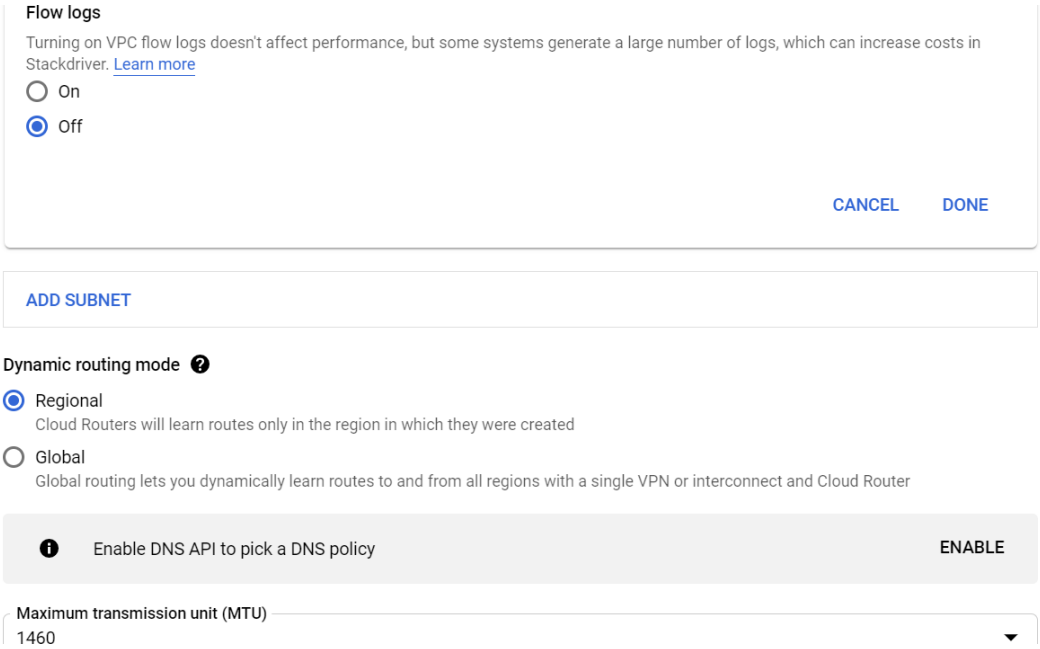
2) **Populate Name, Description and select subnet creation mode as custom. Replace "anil" with your name in the name of VPC.**



3)Populate Subnet name, Subnet description, Region, IP address range. Replace name with your name in subnet.
Chose region as us-central-1 and ip-address-range 10.0.0.0/29



4)Keep rest of things as default value and click create.



5)Verify that Custom vpc is created as below.

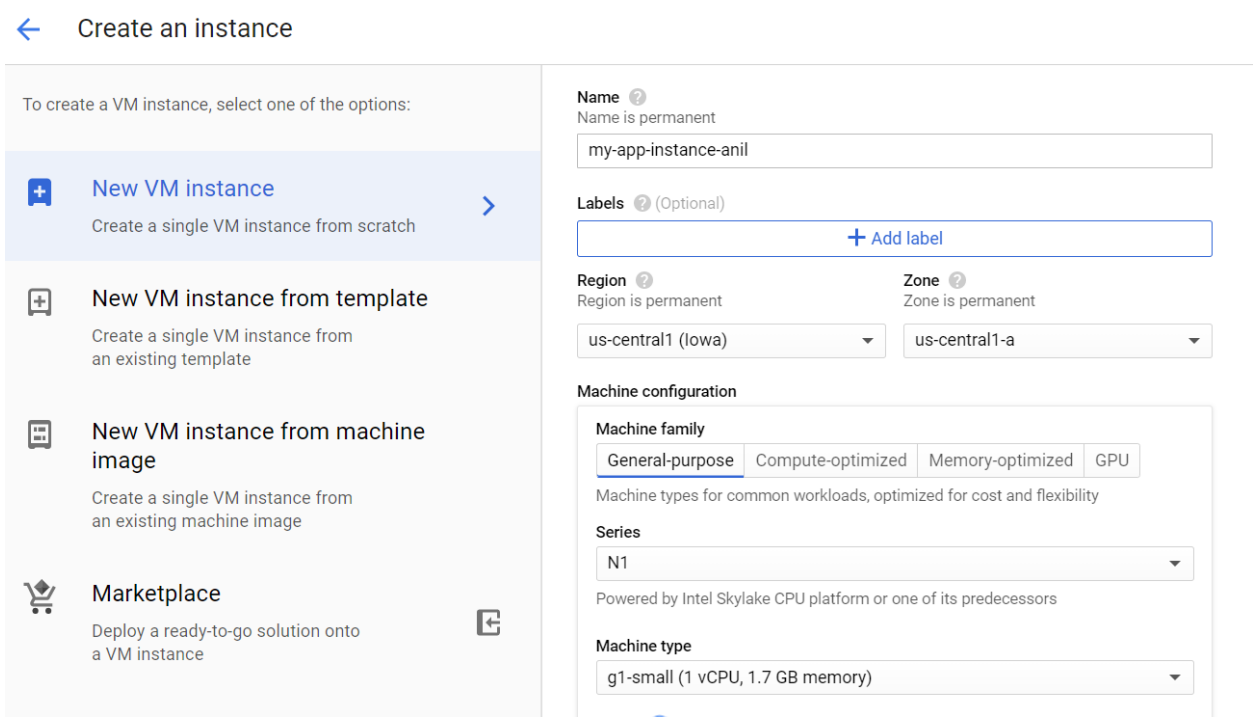
Name	Region	Subnets	MTU	Mode	IP address ranges	Gateways	Firewall Rules	Global
▼ custom-vpc-network-anil		1	1460	Custom			0	Off
	us-central1	subnet-us-central1-anil			10.0.0.0/29	10.0.0.1		

App instance setup

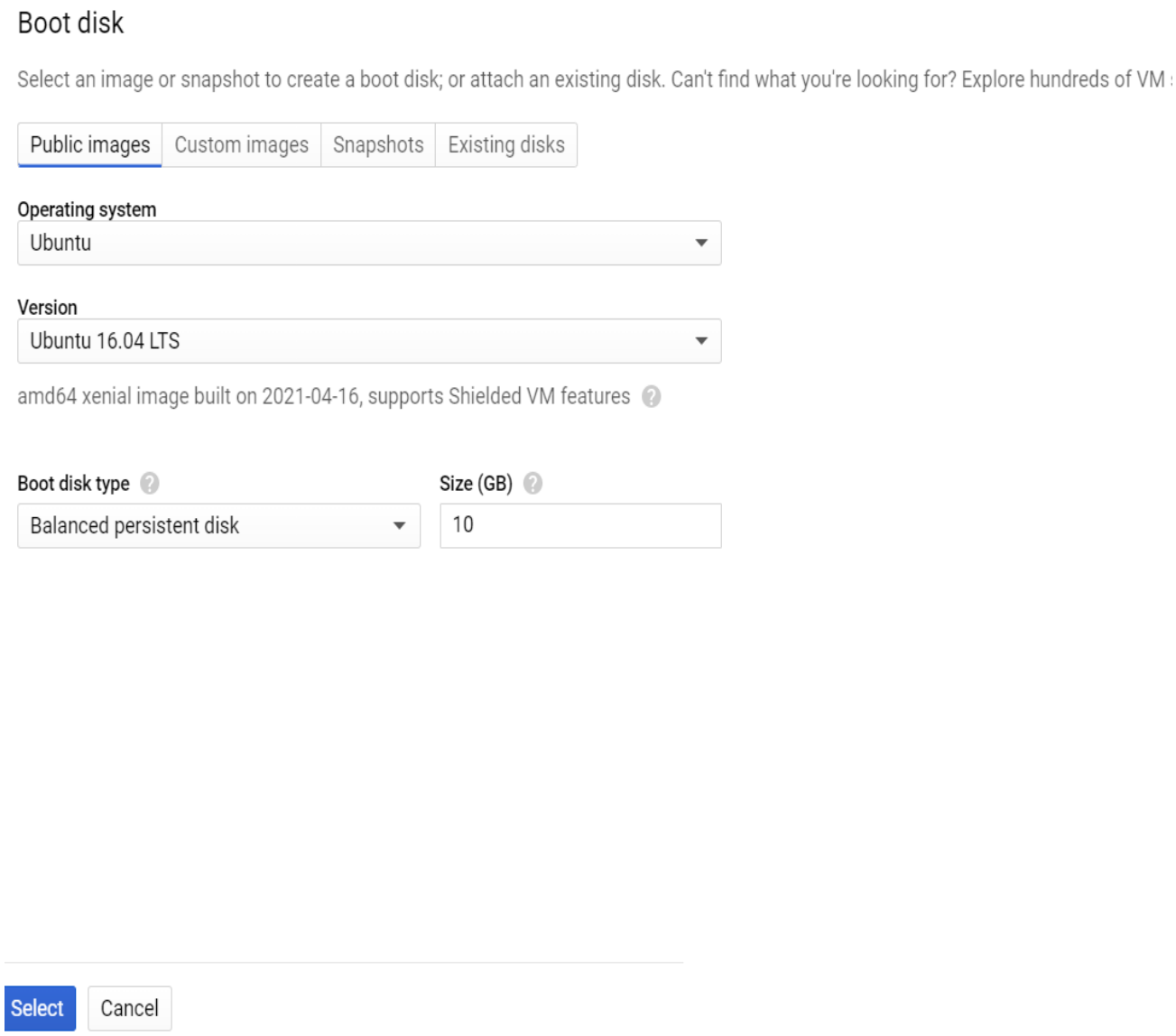
6)we have to now create a VM instance in our custom vpc network .

Populate instance name ,region (us-central1),zone (any zone from us-central 1 region),Series (N1),(machine-type g1-small)

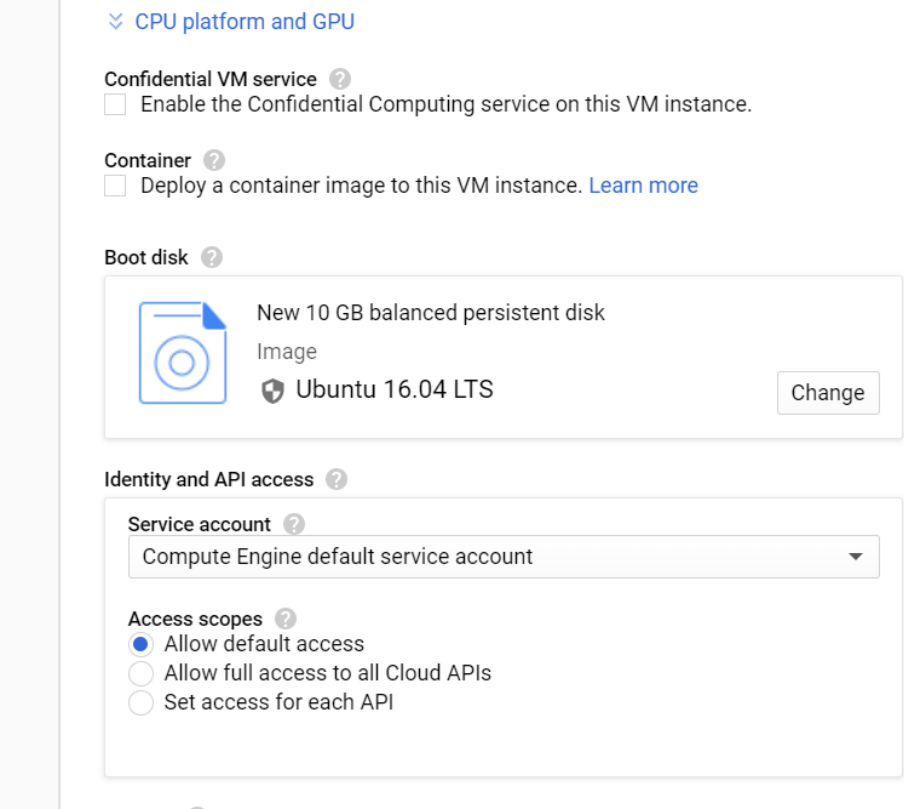
Replace anil with your name in Vm instance name.



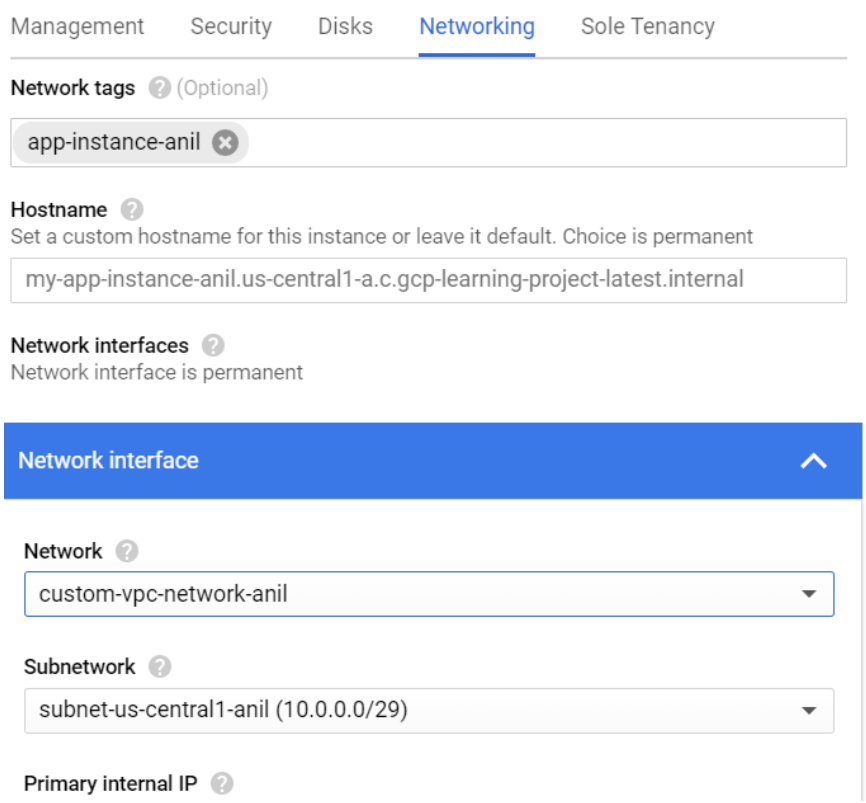
7)select boot disk as ubuntu and version 18.04 LTS



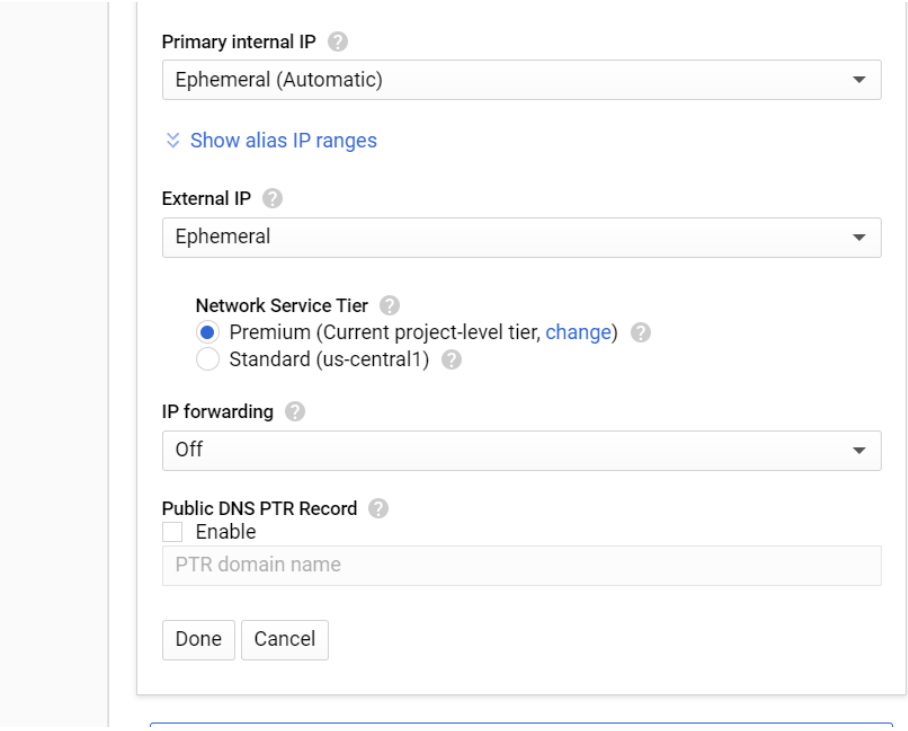
8)Verify the boot disk is selected as ubuntu.



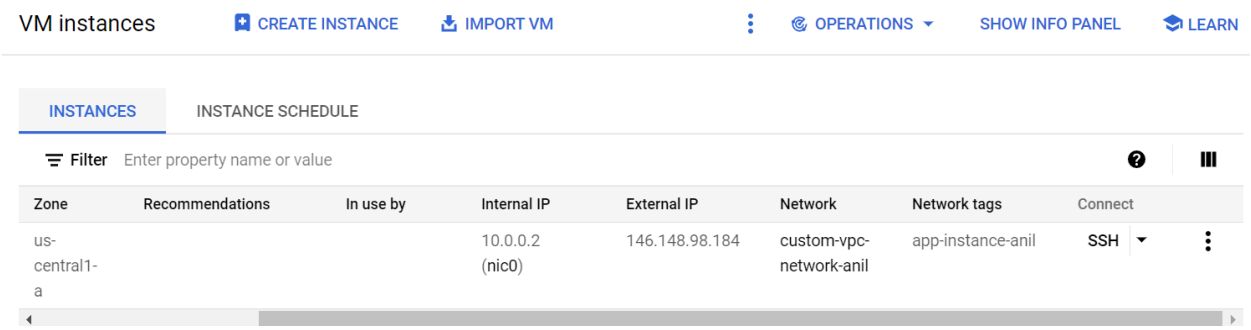
9)Populate network tags (app-instance-anil),network (custom-vpc-network-anil) and subnet (subnet-us-central1-anil) Note put your name where anil is applicable.



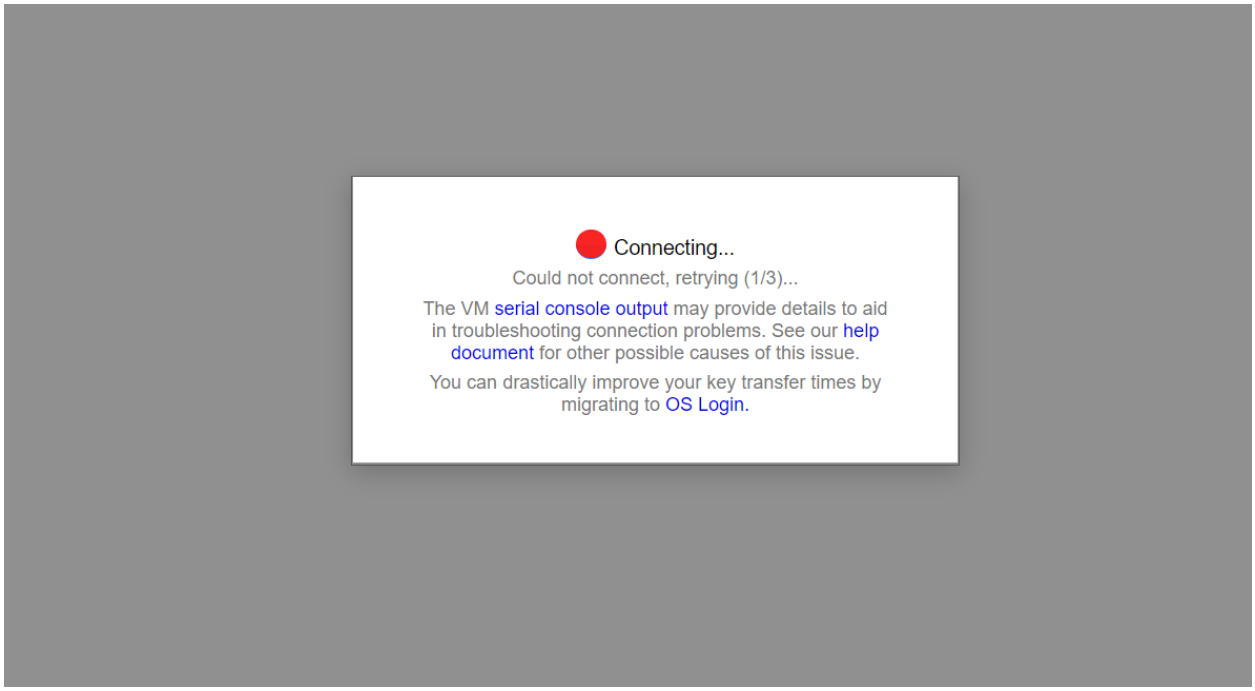
10)Keep rest of the things as default and click create.



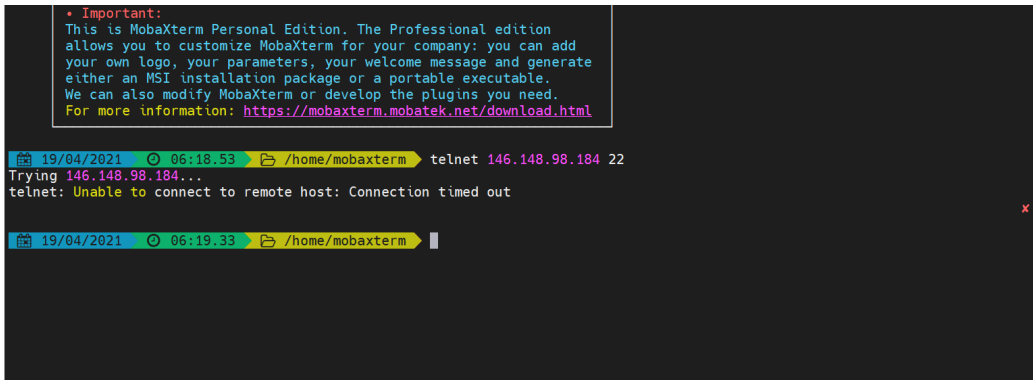
11)Click on SSH to login in to the VM instance .



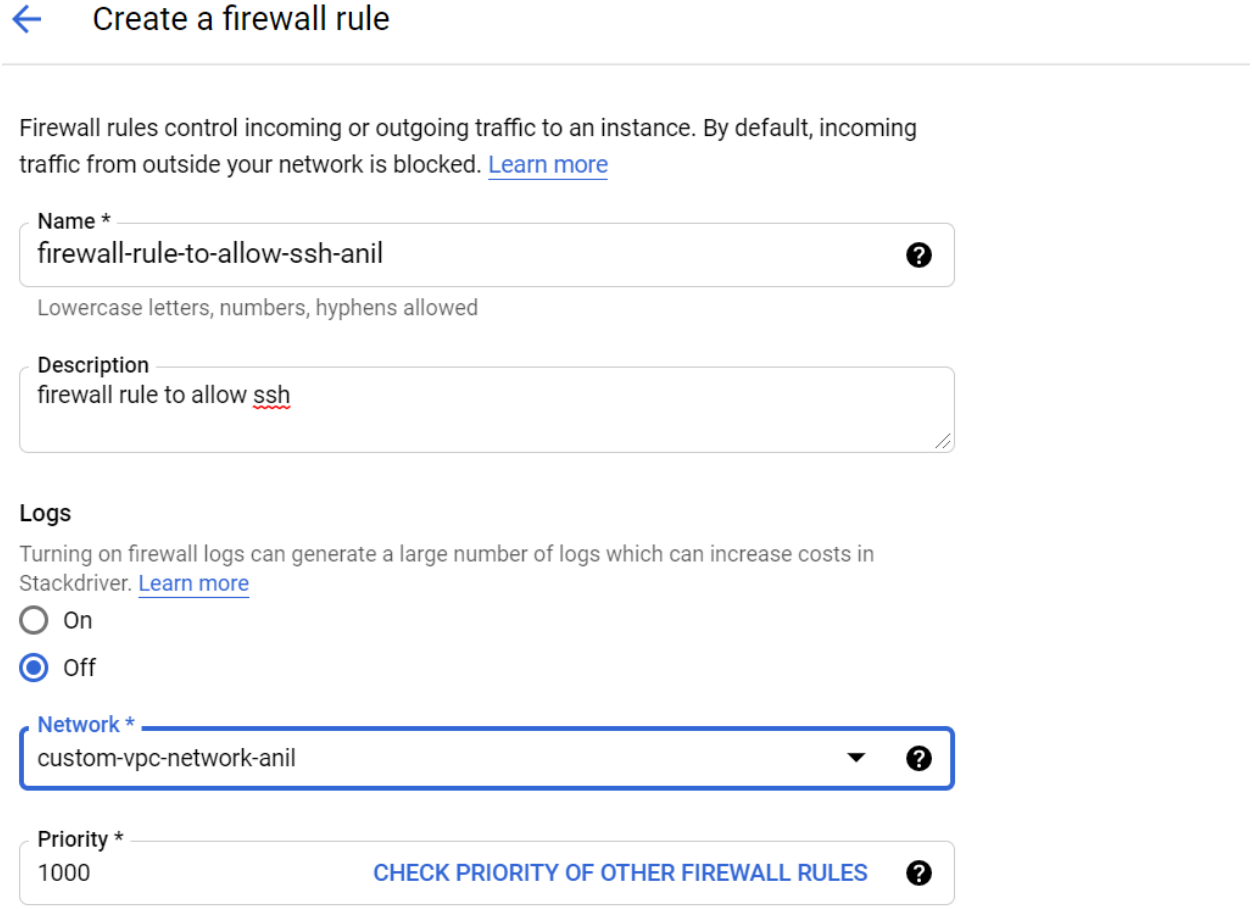
12)Verify that we are not able to do SSH login in VM instance.



13)Verfiy that telnet is also not working from your local machine on port 22 on external ip address of vm instance.



14)Add firewall rule to allow SSH on port 22.
Populate Name,description and network(custom-vpc-network-anil)



15)Specify target tags and source ip filter as below

Direction of traffic

- (●) Ingress
- ( ) Egress

Action on match

- (●) Allow
- ( ) Deny

Targets
Specified target tags

Target tags *
app-instance-anil

Source filter
IP ranges

Source IP ranges *
0.0.0.0/0 for example, 0.0.0.0/0, 192.168.2.0/24

Second source filter
None

16)Specify protocol as tcp and port as 22 and click create

Protocols and ports

- ( ) Allow all
- (●) Specified protocols and ports

[x] tcp : 22

[ ] udp : all

[ ] Other protocols

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

DISABLE RULE

CREATE CANCEL

17)Verify the firewall rule is now showing app-instance in its applicable instance list when we click on our firewall rule.

Insights

None

Hit count monitoring

–

Applicable to instances

The following table shows only the VM instances that you have permission to view. It also does not show any App Engine flexible environment instances.

Filter Filter by instance name, project or subnetwork

Name	Subnetwork	Internal IP	Tags	Service accounts	Project	Labels
my-app-instance-anil	subnet-us-central1-anil	10.0.0.2	app-ins...	302020575003-compute@developer.gse...	gcp-learning-project-latest	

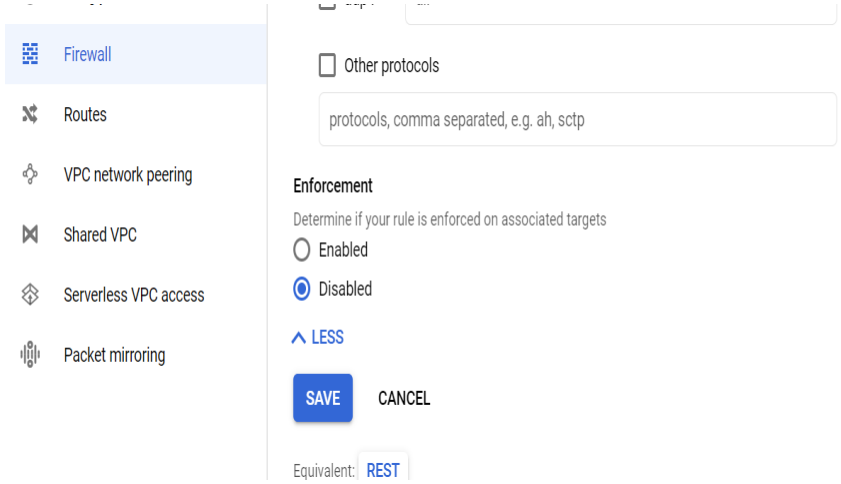
18)Verify the telnet is now allowing to go through on port 22 on external ip address

```
19/04/2021  06:19.33  /home/mobaxterm  telnet 146.148.98.184 22
Trying 146.148.98.184...
Connected to 146.148.98.184.
Escape character is '^]'.
SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.10

telnet> quit
Connection closed.

19/04/2021  06:28.53  /home/mobaxterm
```

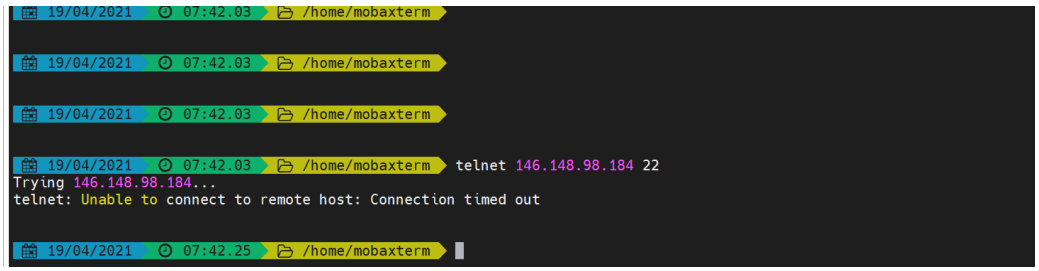
Note To come out of telnet session press (ctrl +]) key from keyboard. then type quit

19)Verify now we are able to do ssh in the vm instance after adding the firewall rule for ssh 22.At this point we shall be able to login to VM instance now.

20)Disable the firewall rule for ssh and verify that after disabling firewall rule we are not able to login to ssh and also telnet is not able to connect on port 22.



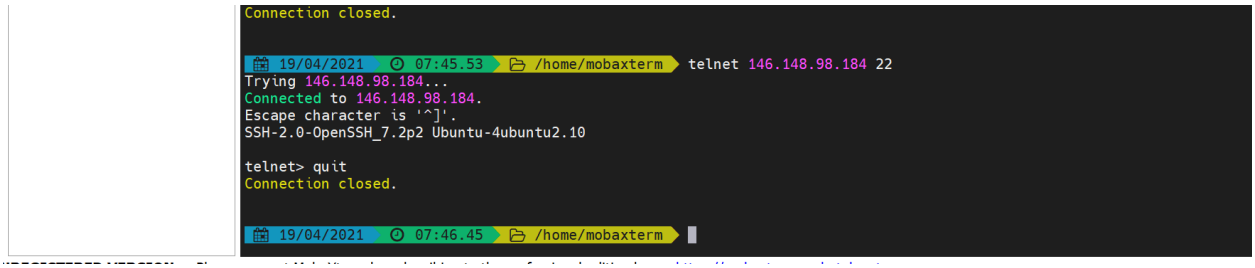
Telnet on 22



verify that we are not able to do SSH using google cloud console.

21)Enable the firewall rule (firewall-rule-to-allow-ssh-anil) to allow SSH access

Verify telnet is working now fine and we are also do login to vm instance using SSH in google console.



22)Upload the file my_web_app.py in app instance (my_web_app.py)
23)Starting app web server on app instance.
sudo apt update
sudo apt install python3-pip
sudo pip3 install flask
sudo apt-get install libpq-dev
sudo pip3 install psycopg2-binary
python3 my_web_app.py

24)Verfiy server is listening on port 3000,open another ssh session of virtual machine an do netstat on port 3000
netstat -an |grep 3000

```
tcp        0      0 0.0.0.0:3000            0.0.0.0:*
LISTEN
```

25)Verify that telnet is not going through on port 3000
telnet 146.148.98.184 3000
Note 146.148.98.184 is external ip address of my virtual machine.

26)Verify in the browser we are not able to access on port 3000
http://146.148.98.184:3000/
Note 146.148.98.184 is external ip address of my virtual app instance

27)Create another firewall rule to allow app access on port 3000.

Create a firewall rule

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

Name *
firewall-rule-to-allow-app-access-anil

Lowercase letters, numbers, hyphens allowed

Description
firewall rule to allow app access

Logs

Turning on firewall logs can generate a large number of logs which can increase costs in Stackdriver. Learn more

- On
- Off

Network *
custom-vpc-network-anil

Priority *
1000 CHECK PRIORITY OF OTHER FIREWALL RULES

Priority can be 0 - 65535

Targets
Specified target tags

Target tags *
app-instance-anil

Source filter
IP ranges

Source IP ranges *
0.0.0.0/0 for example, 0.0.0.0/0, 192.168.2.0/24

Second source filter
None

Protocols and ports

- Allow all
- Specified protocols and ports

[x] tcp : 3000

28)After creating the firewall for port 3000,hit browser with URL
http://146.148.98.184:3000/
output shall be welcome to custom network of gcp!

29)Verify that we are also able to do telnet on port 3000
telnet 146.148.98.184 3000

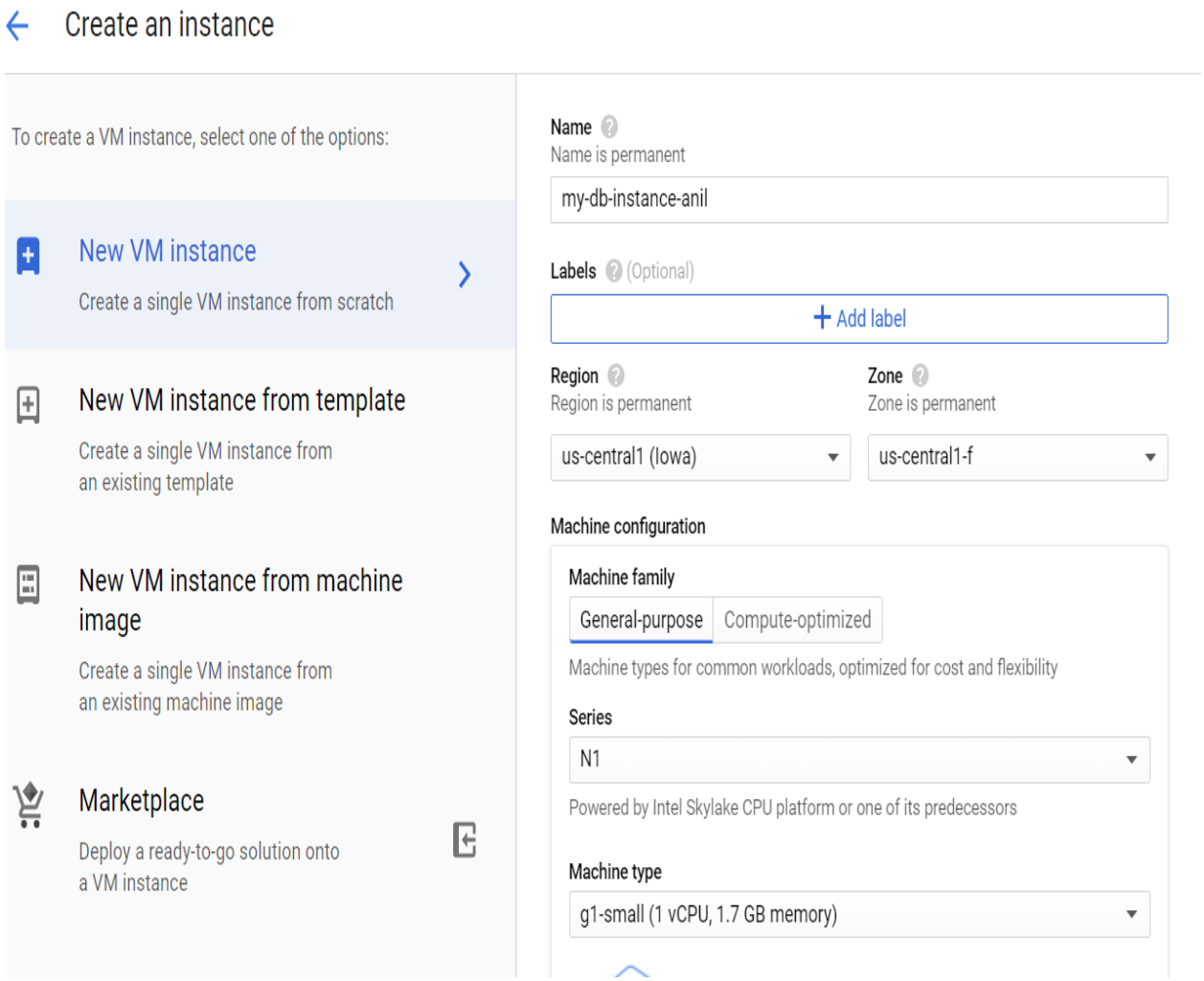
30)Disable the firewall rule (firewall-rule-to-allow-app-access-anil) for port 3000 and verify that telnet is not going through and browser is also not giving the output now.

telnet 146.148.98.184 3000
http://146.148.98.184:3000/ (from browser)

31)Enable the firewall rule (firewall-rule-to-allow-app-access-anil) for port 3000 and verify that telnet is going through and browser is also giving the output now.

DB instance Set up

32)We have to create DB instance as below. Ensure that we have ubuntu and network tag properly populated as below.



Container

Deploy a container image to this VM instance. Learn more

Boot disk

New 10 GB balanced persistent disk

Image

Ubuntu 16.04 LTS

Change

Identity and API access

Service account

Compute Engine default service account

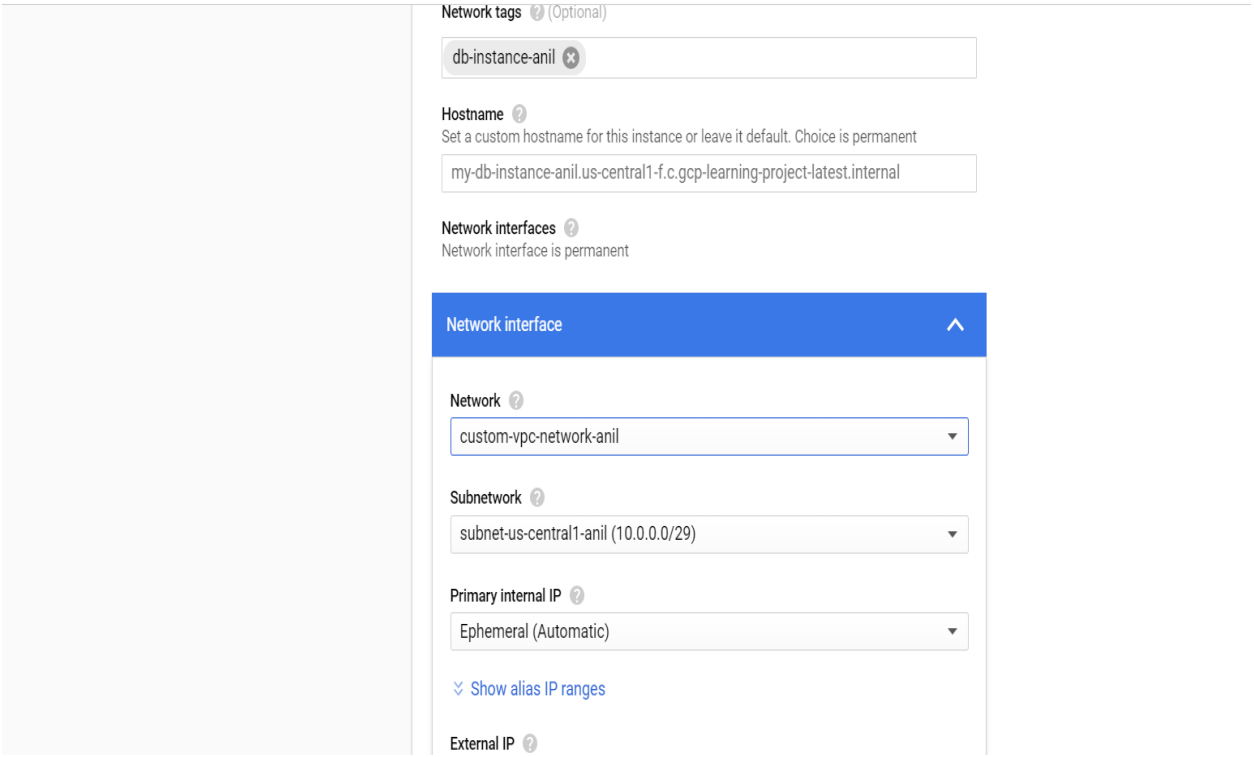
Access scopes

- (●) Allow default access
- ( ) Allow full access to all Cloud APIs
- ( ) Set access for each API

Firewall

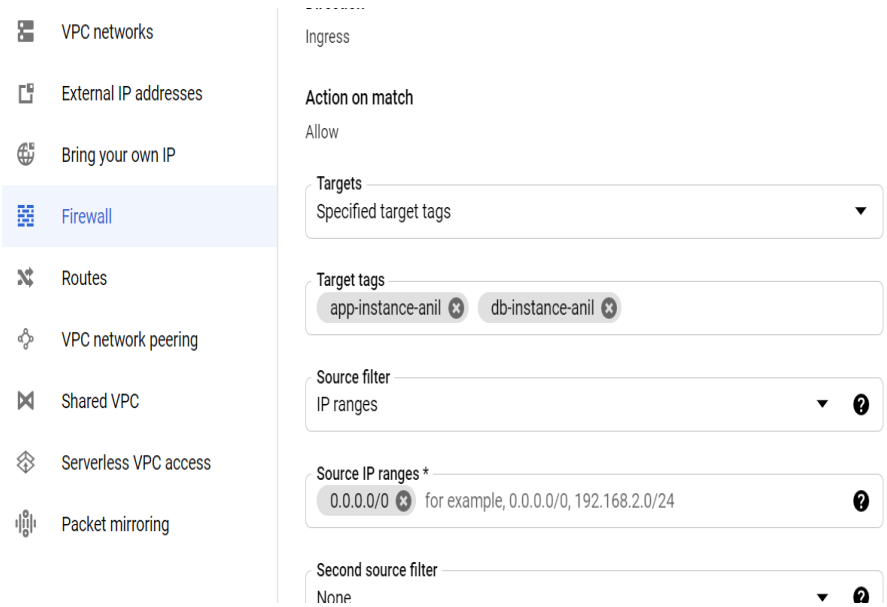
Add tags and firewall rules to allow specific network traffic from the Internet

- [ ] Allow HTTP traffic
- [ ] Allow HTTPS traffic



33)Verify that ssh is not working on db instance and also telnet is not working on db instance.
telnet 35.232.61.234 22
where 35.232.61.234 is external ip address of my db virtual machine.

34)Modify firewall rule firewall-rule-to-allow-ssh-anil to add target tag for db instance as below



35)Verify that we are able to login via ssh on db instance now and also able to do telnet now on db instance external ip address.
telnet 35.232.61.234 22

36)we need to install postgres on db instance machine.
sudo apt-get update
sudo apt-get install postgresql

37)Create table employee in postgres databse and change password
sudo -u postgres psql
create table employee(id int,emp_name text,emp_organization text,emp_salary int,emp_age int);
alter user postgres password '123456';

Note to come out of postgres shell use \q and then enter

38)Do netstat on db instance on port 5432
netstat -an |grep 5432

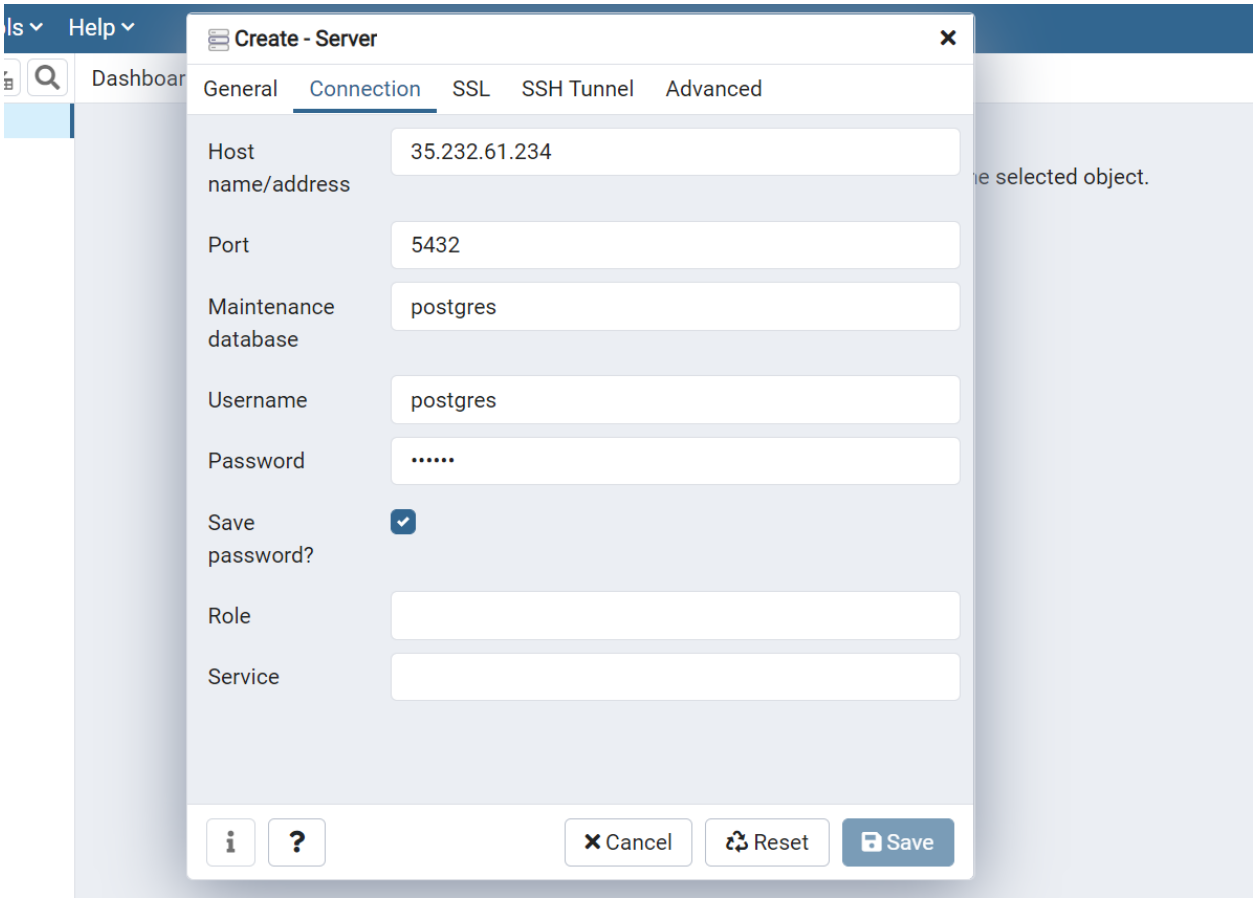
output shall contain
tcp 0 0 127.0.0.1:5432 0.0.0.0:* LISTEN

39) To check postgresql process is running
ps -eaf |grep postgresql
output shall be
postgres 4526 1 0 15:40 ? 00:00:00
/usr/lib/postgresql/9.5/bin/postgres -D /var/lib/postgresql/9.5/main -c
config_file=/etc/postgresql/9.5/main/postgresql.conf

40)On db-instance session do telnet localhost on 5432 ,we shall get connected
telnet localhost 5432

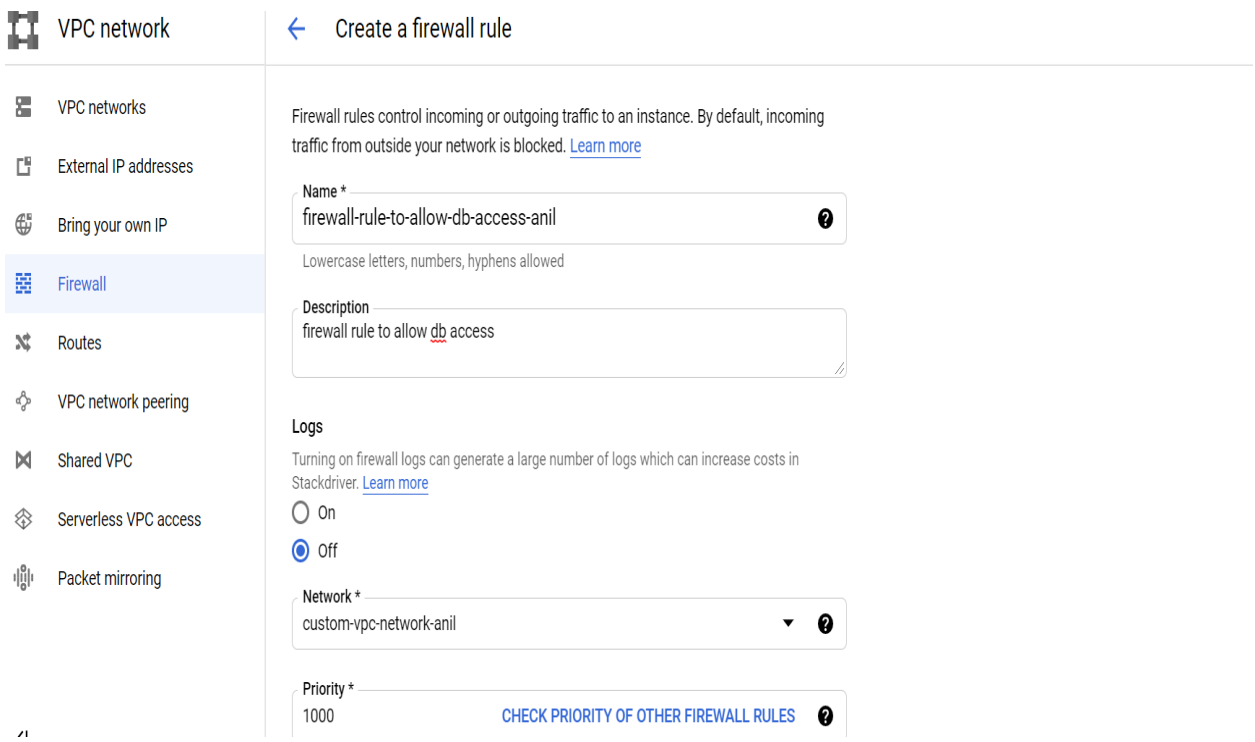
41)Verify that From your local machine we don't connect on telnet on port 5432 by using external ip address of db instance
telnet 35.232.61.234 5432

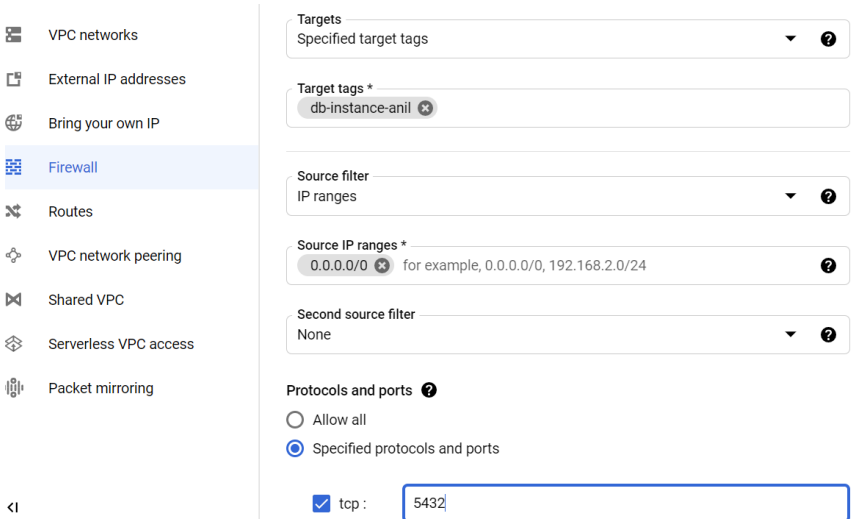
42)Go to pg admin and create new server and give Name cloud-server.
In connection tab give following details



Note IP address mentioned above is the external ip address of Database server.
We shall not be able to connect in pgadmin as well.

43)Create a firewall rule to allow access to 5432 for db-instance.
Ensure that we select the proper network tag and port number.

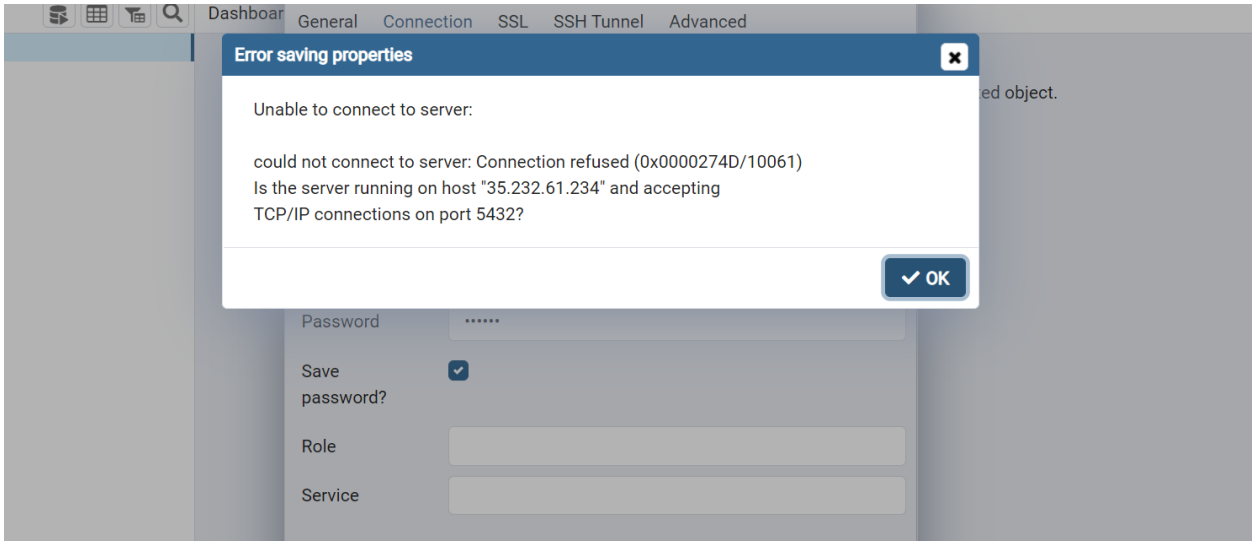




44)Verify that firewall rule (firewall-rule-to-allow-db-access-anil) has db instance as applicable instances.

45)Try to do telnet again on port 5432 from local machine
telnet 35.232.61.234 5432
Telnet shall not be able to go through .

46)Go to pg admin and again do save the connection details to see we are able to connect to pg admin or not.



We shall not be able to connect to database postgres.
The reason is that postgres by default is not allowing connection from remote machine. By default it allows connection from client running on same machine where postgres is running.

47)Setup for postgres to allow remote connections.(2 file changes)
On db instance Go to path /etc/postgresql/10/main
File 1 change postgresql.conf
grep "listen_address" postgresql.conf
output shall be
#listen_addresses = 'localhost' # what IP address(es) to listen on;

We have to uncomment above line and change localhost to * as below
listen_address = '*'

use below vi command to do the changes
sudo vi postgresql.conf

File 2 change pg_hba.conf

add below line at the end of file (shift G,$,a,enter)
sudo vi pg_hba.conf
host all all 0.0.0.0/0 trust

Once both files are changed ,restart the postgresql service as below
sudo service postgresql restart

48)To Verify all the changes are properly done at postgresql side ,do following command
netstat -an |grep 5432
output shall have
tcp 0 0 0.0.0.0:5432 0.0.0.0:* LISTEN

Note in output of above command if we have 127.0.0.1:5432,then changes are not done properly at postgres sql side

49)Do telnet from local machine on port 5432,we shall now be able to connect on port 5432

50)Check in pg admin we shall now able to connect to cloud via pg admin as well.

51)Insert five employee records in employee table from pg admin now. Below is first insert command.
insert into
employee(id,emp_name,emp_organization,emp_salary,emp_age) values (123,'anil','agilitics',1000,23);

52)Read all records from employee table in pg admin

Select * from employee;
53)Delete one employee record from employee table based on id.
select * from employee;

54)Read all records back from employee table in pg admin
Select * from employee;
It shall now have only four records

55)Reading all database records from my_web_app.py running on local machine
Change ip of host in method create_connection() to external ip address of db instance. (start the application again python3 my_web_app.py)

Hit below url from broswer
http://<external_ip_app>:3000/read

we shall see all the records from employee table that we inserted using pg admin.

55)Insert one more record in employee table using pg admin and refresh your browser (http://external_ip_app:3000/read) ,we shall now see new record as well.

56)Insert one more record from your application running on localmachine by using below end point
http://external_ip_app:3000/write?id=666&name=rahul&age=99
Here we are adding new record via application code with id 666,name = Rahul and age=99

57)Verify that new record got added in the cloud database
From browser: http://external_ip_app:3000/read
From pg admin: select * from employee;

57)Not required
Disable the firewall rule (firewall-rule-to-allow-db-access-anil) and verify that telnet from local,read and write end points from local application will stop working and also pg admin.

Note pg admin keeps a connection ,try disconnect and then try to connect back.

58)Application running on cloud will connect to database running on cloud (we need to use internal IP)

In SSH session of application instance, do a ping to internal IP of db instance
ping 10.0.0.3
Here 10.0.0.3 is internal ip of db instance.

There shall not be any packet transmission from app instance to db instance via ping.

59)create a firewall rule to allow only icmp from appisntance to db instance (ensure that we put proper network tags.target is db-instance-anil and source is app-instance-anil)

60)verify that ping is working now from source machine (app instance) to db instance.

Disable the existing firewall rule for db on port 5432.

From machine app instance Do telnet on internal ip of db on port 5432 .
It shall not work.

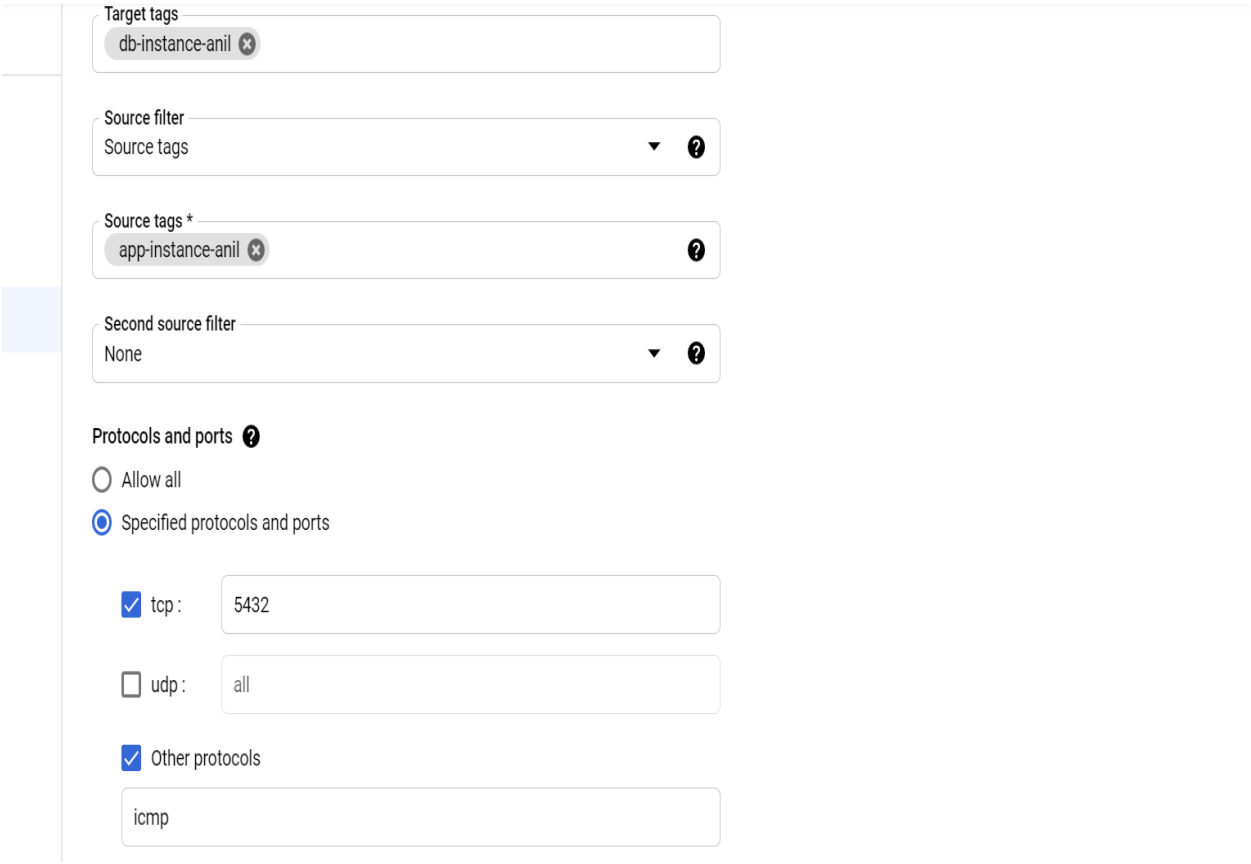
61)Modify the existing firewall from app-instance-to-db-instance to allow access on port 5432.

Verify telnet is working from app instance to db instance now

62)change the ip address of host to internal ip address of db instance and hit the url
http://35.192.0.66:3000/read

we shall see all the records are coming back.

Don't attempt below one section
59)Edit the firewall rule (firewall-rule-to-allow-db-access-anil) to allow app-instance to ping to db-instance .
Ensure that we are having source tag as app-instance-anil and icmp is mentioned in other protocol.(enable the firewall rule)



60)Verify that ping is working from app instance to db instance on internal ip address.

61)Verify that we are able to do telnet from app instance to db instance on port 5432 on internal ip address
telnet 10.0.0.3 5432
Here 10.0.0.3 is internal ip address of db instance.

62)Edit the firewall rule (firewall-rule-to-allow-db-access-anil) and remove icmp and unselect other protocol .Keep rest of things as it is

Ping from app instance to db instance shall stop working
Telnet from app instance to db instance shall still happen as it is tcp connection.

63)Point your cloud application code to postgres running on cloud. Change the host ip address in my_web_app.py to internal ip of db instance.

64)Verify application running on cloud is pointing to postgres running on cloud
From browser
http://146.148.98.184:3000/read
http://146.148.98.184:3000/write?id=007&name=bond&age=999
http://146.148.98.184:3000/read

Final read shall have bond entry as well

65)Disable the firewall rule (firewall-rule-to-allow-db-access-anil)

66)Verify below end points stop working
http://146.148.98.184:3000/read