How to setup jupyter Notebook on Google Cloud Platform

Sinan Artun · Follow

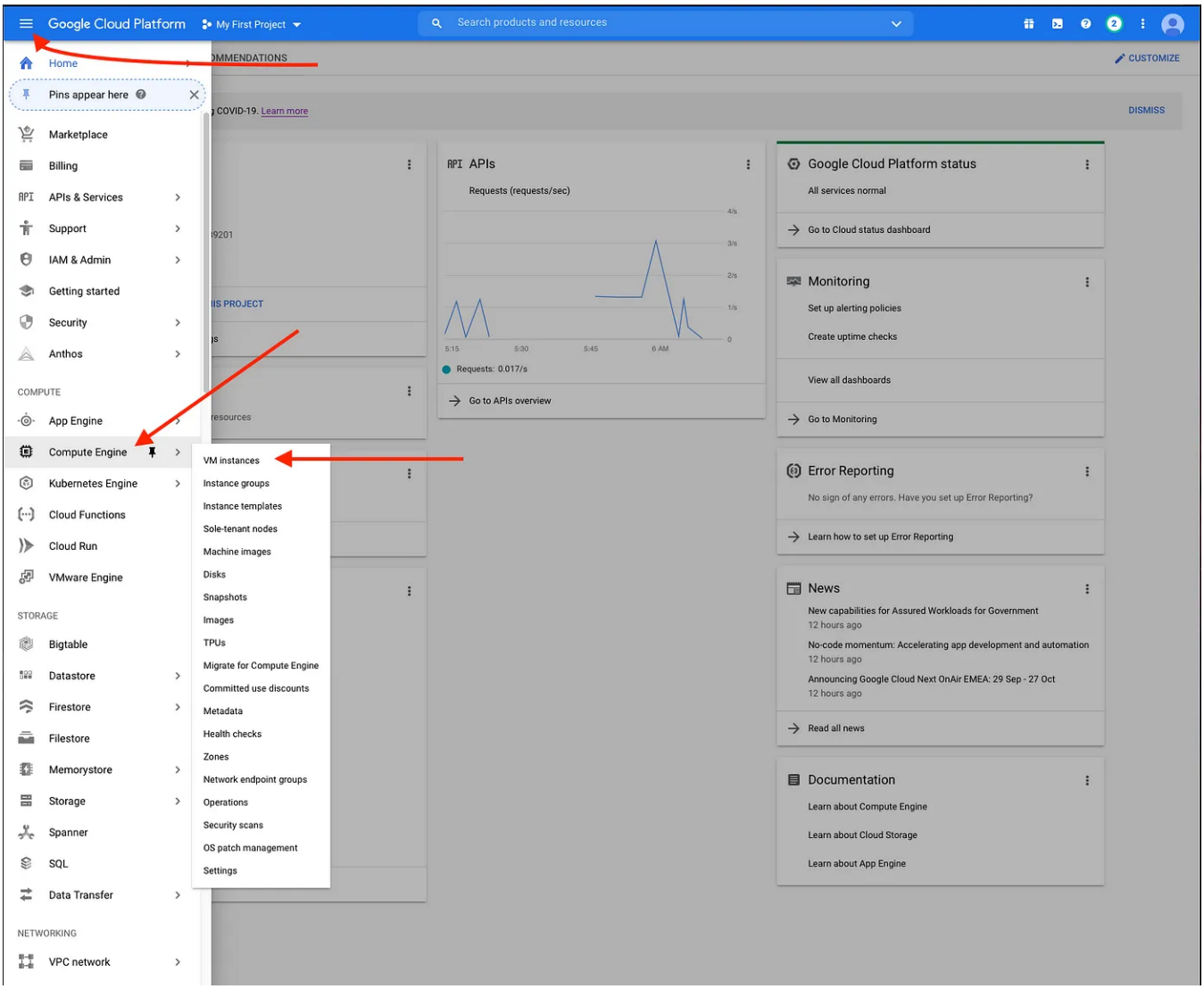
5 min read · Sep 13, 2020

Listen Share

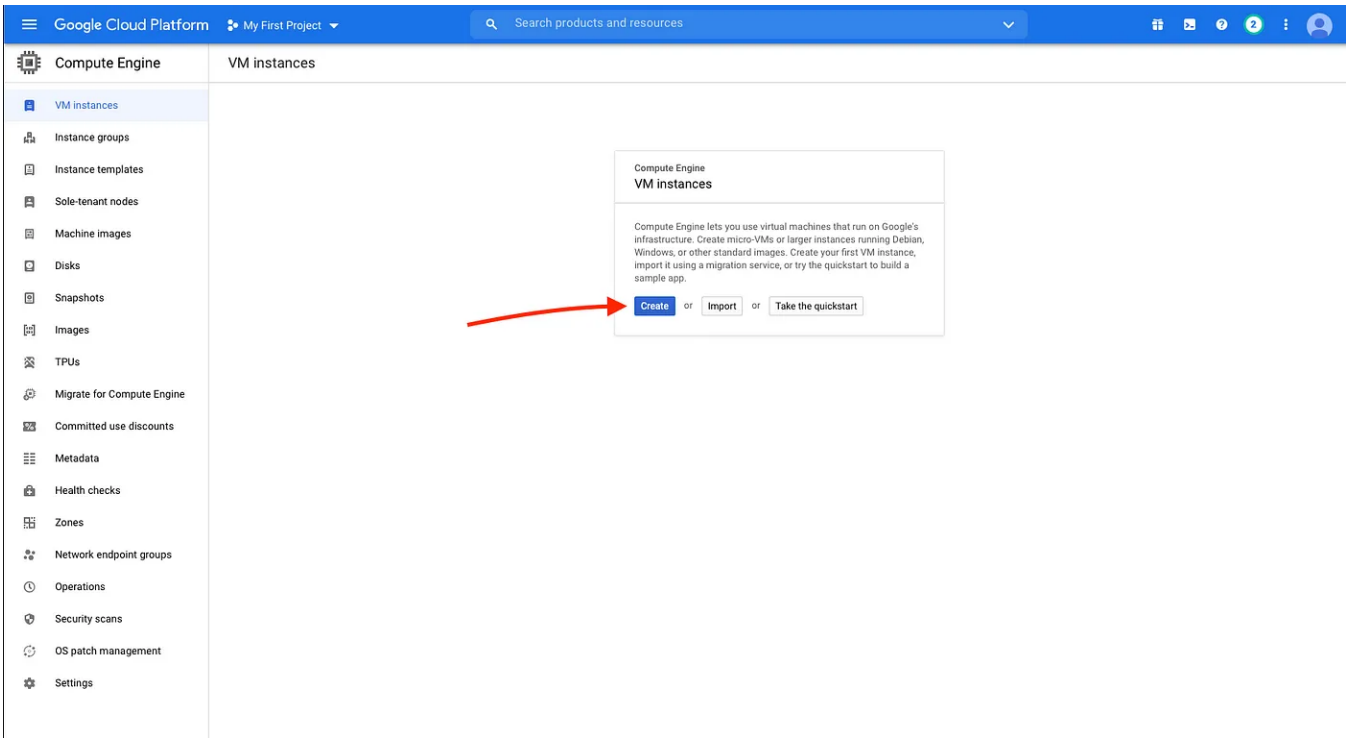
Step 1: Add a VM instance

On Google Cloud Platform Home page Click

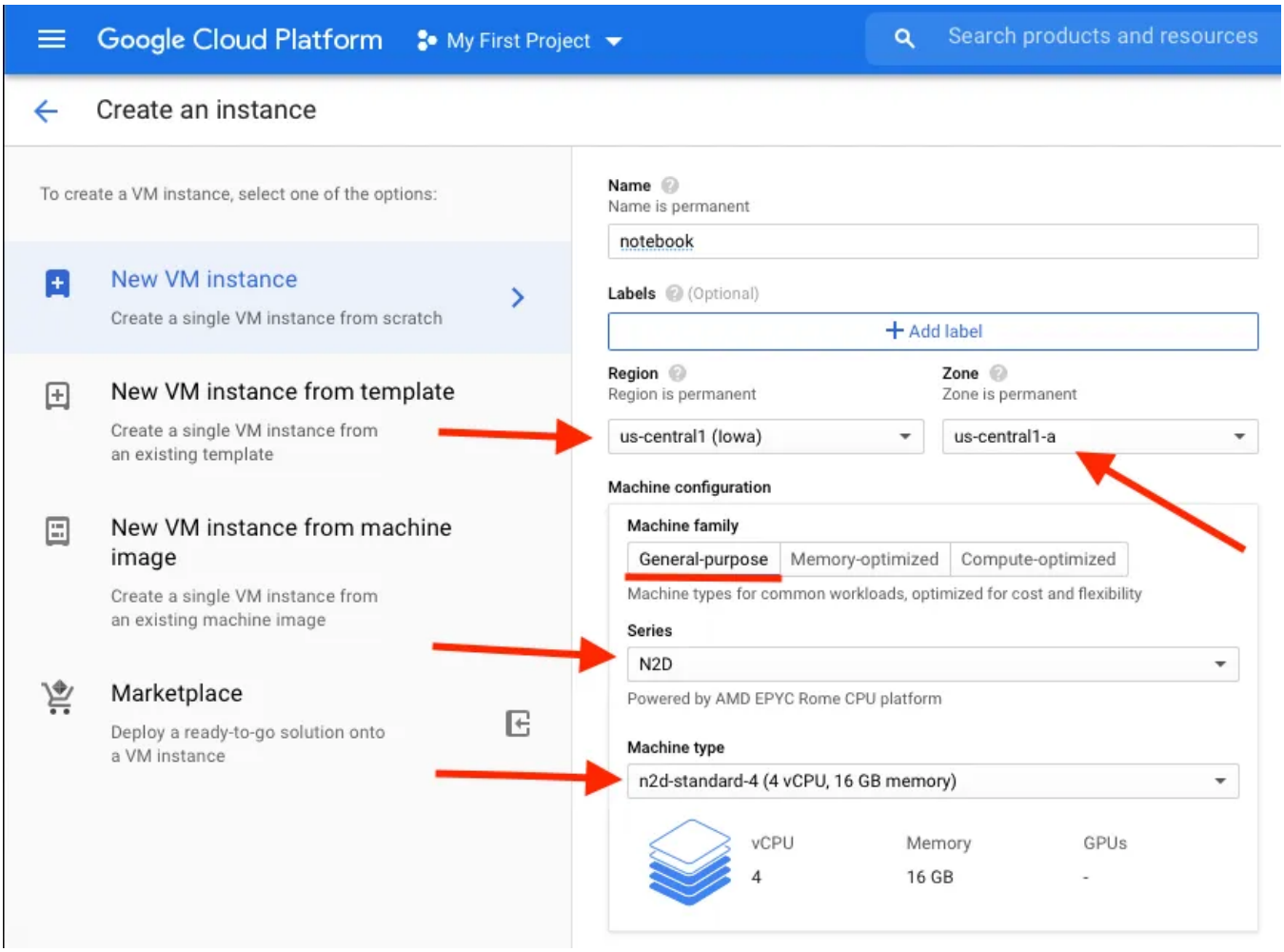
Compute Engine → VM instances



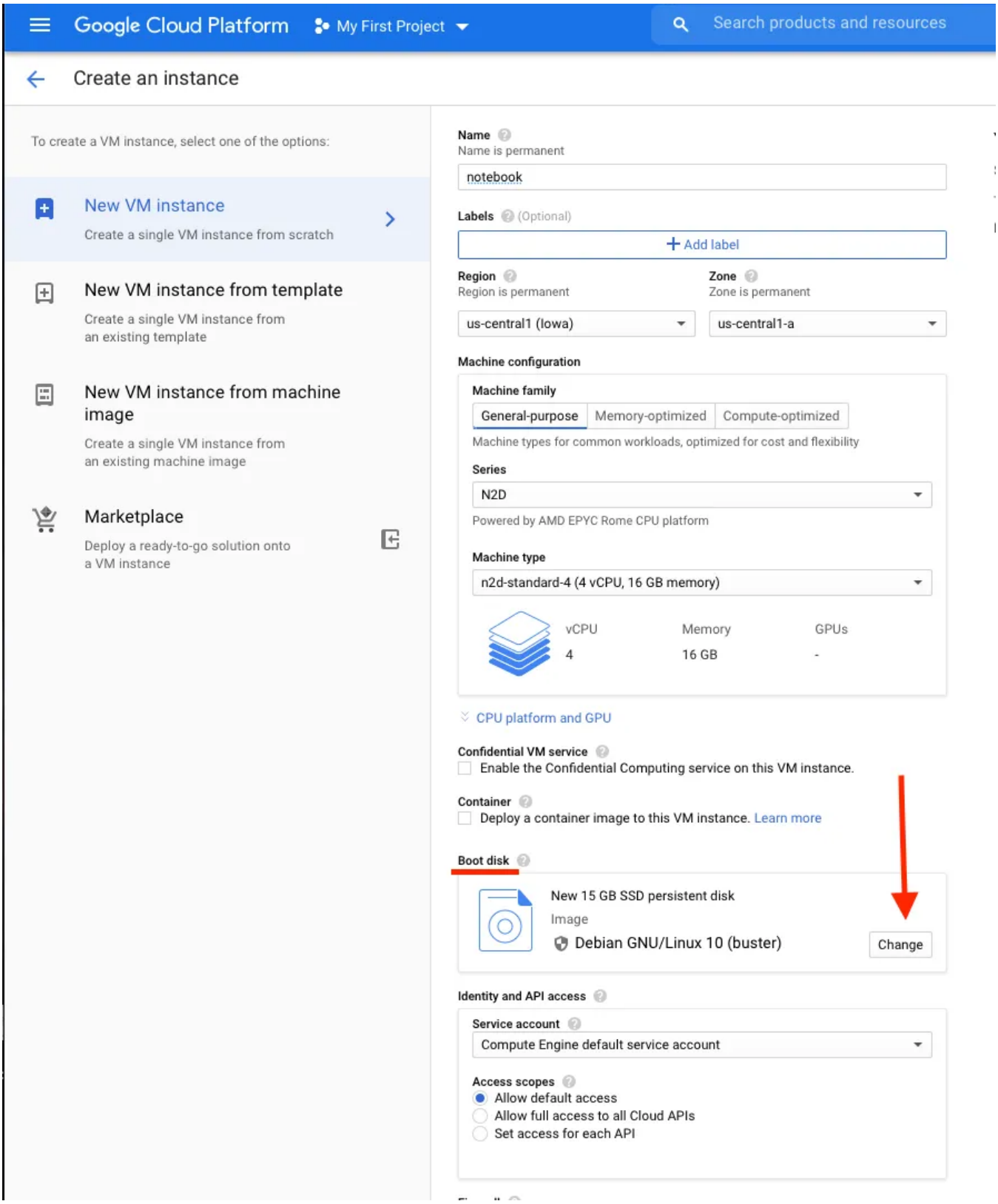
Click Create button



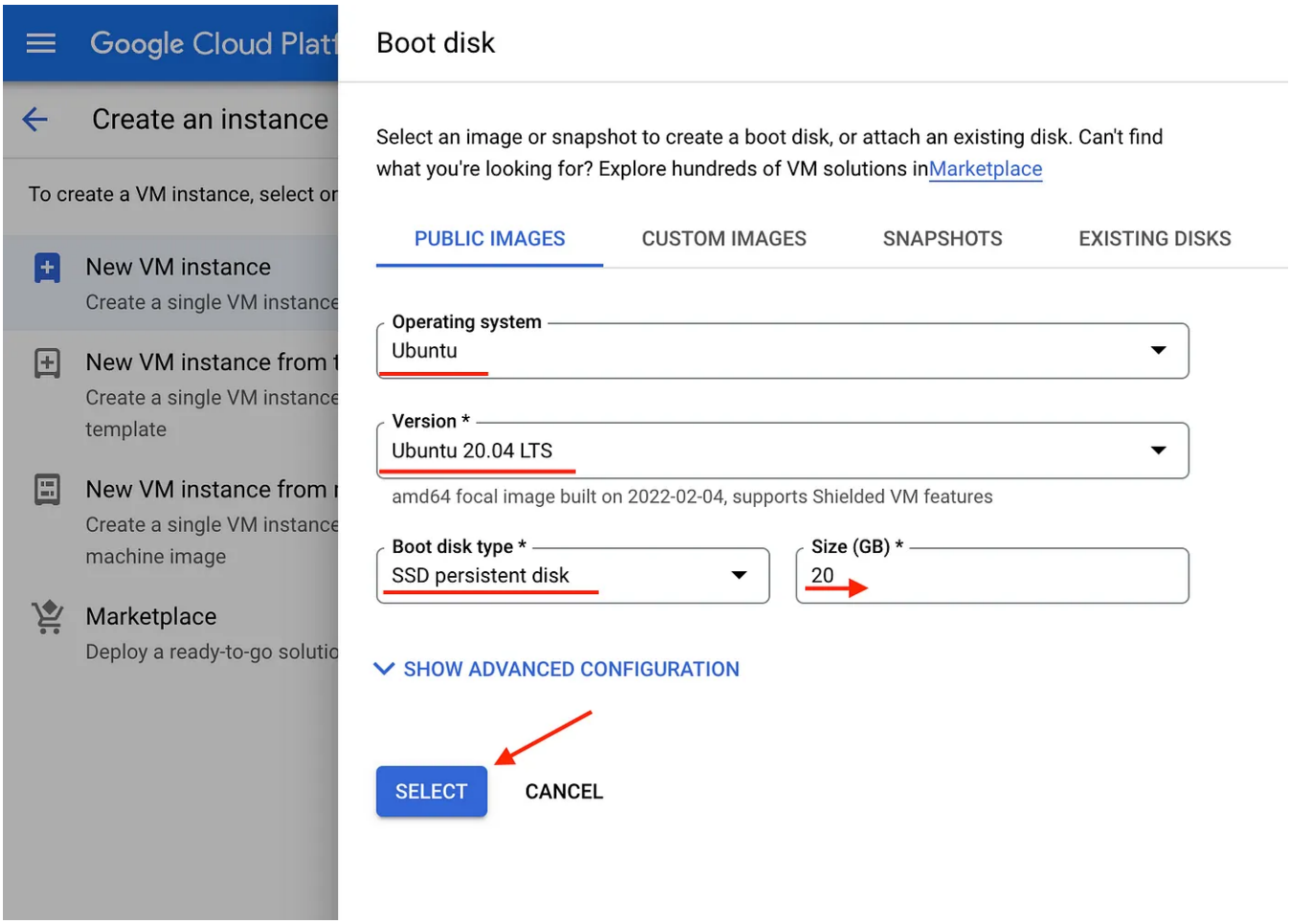
firstly give your instance a name I write “notebook” for this instance. If you are an AMD fan select us-central1 (Iowa) to be able to choose the N2D cpu series, for this project I really need some compute power so I choose **n2d-standard-4** which has 4 cores and 16 GB memory



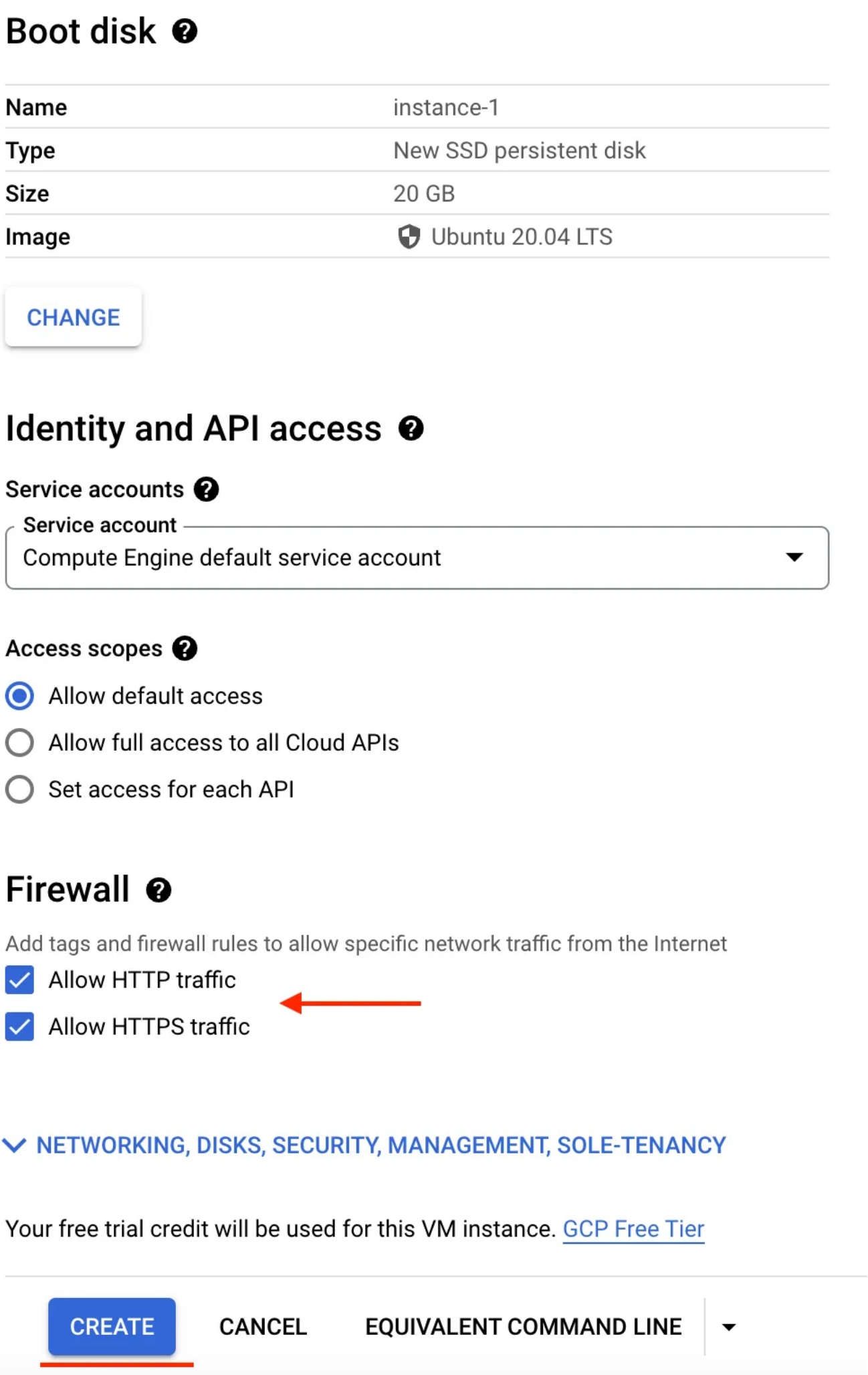
on Boot disk part click the **change** button



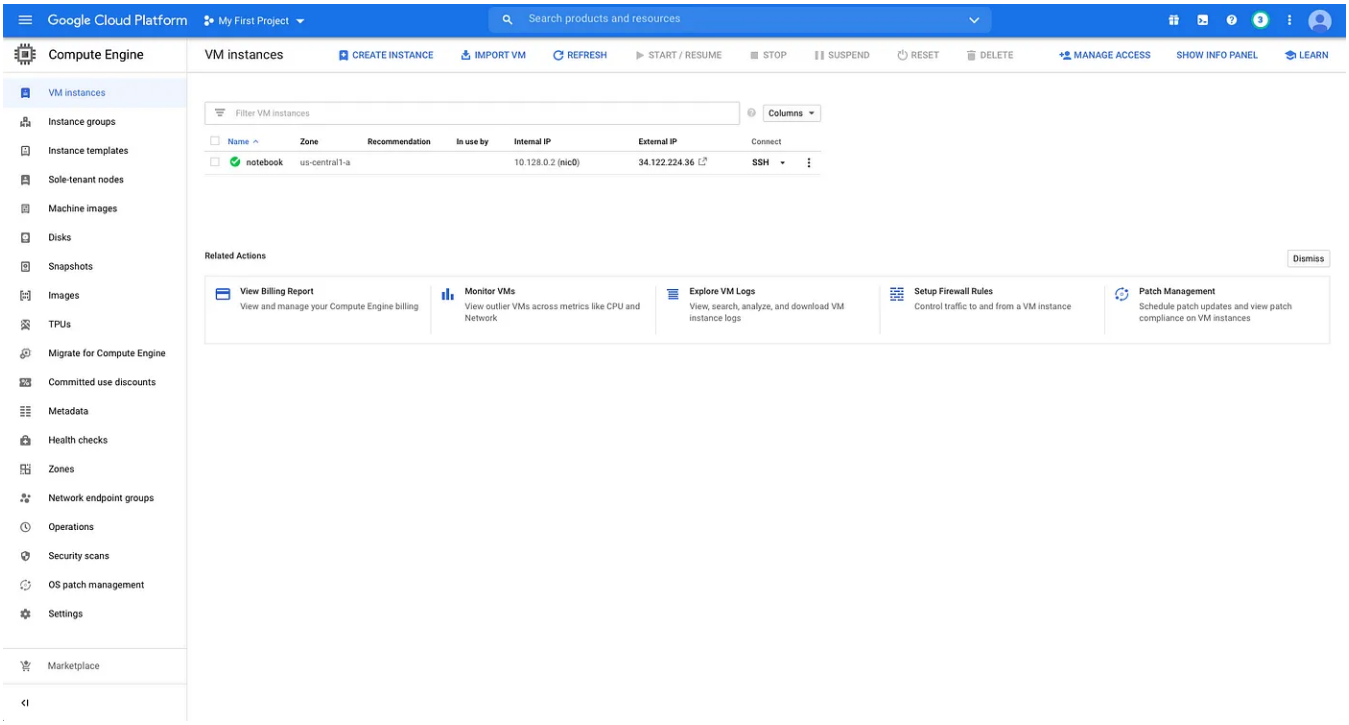
for this project, I want to select Ubuntu 20.04 LTS and SSD persistent disk with 20GB volume.



on Firewall section select **Allow HTTP traffic** and **Allow HTTPS traffic**
You are ready to go, click the create button


Boot disk
Name
instance-1
Type
New SSD persistent disk
Size
20 GB
Image
Ubuntu 20.04 LTS
CHANGE
Identity and API access
Service accounts
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
NETWORKING, DISKS, SECURITY, MANAGEMENT, SOLE-TENANCY
Your free trial credit will be used for this VM instance. GCP Free Tier
CREATE
CANCEL
EQUIVALENT COMMAND LINE

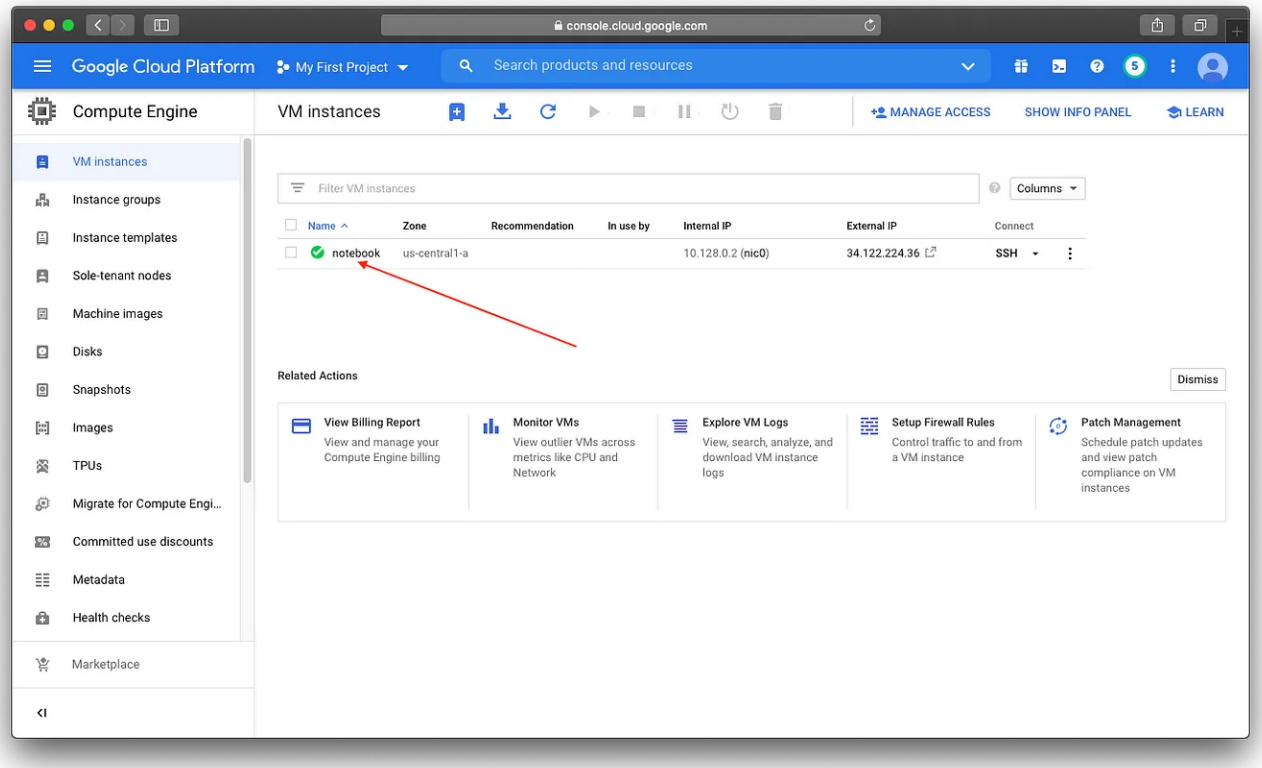
After your VM instance setup finished you will be redirected to the VM instances page which looks like this



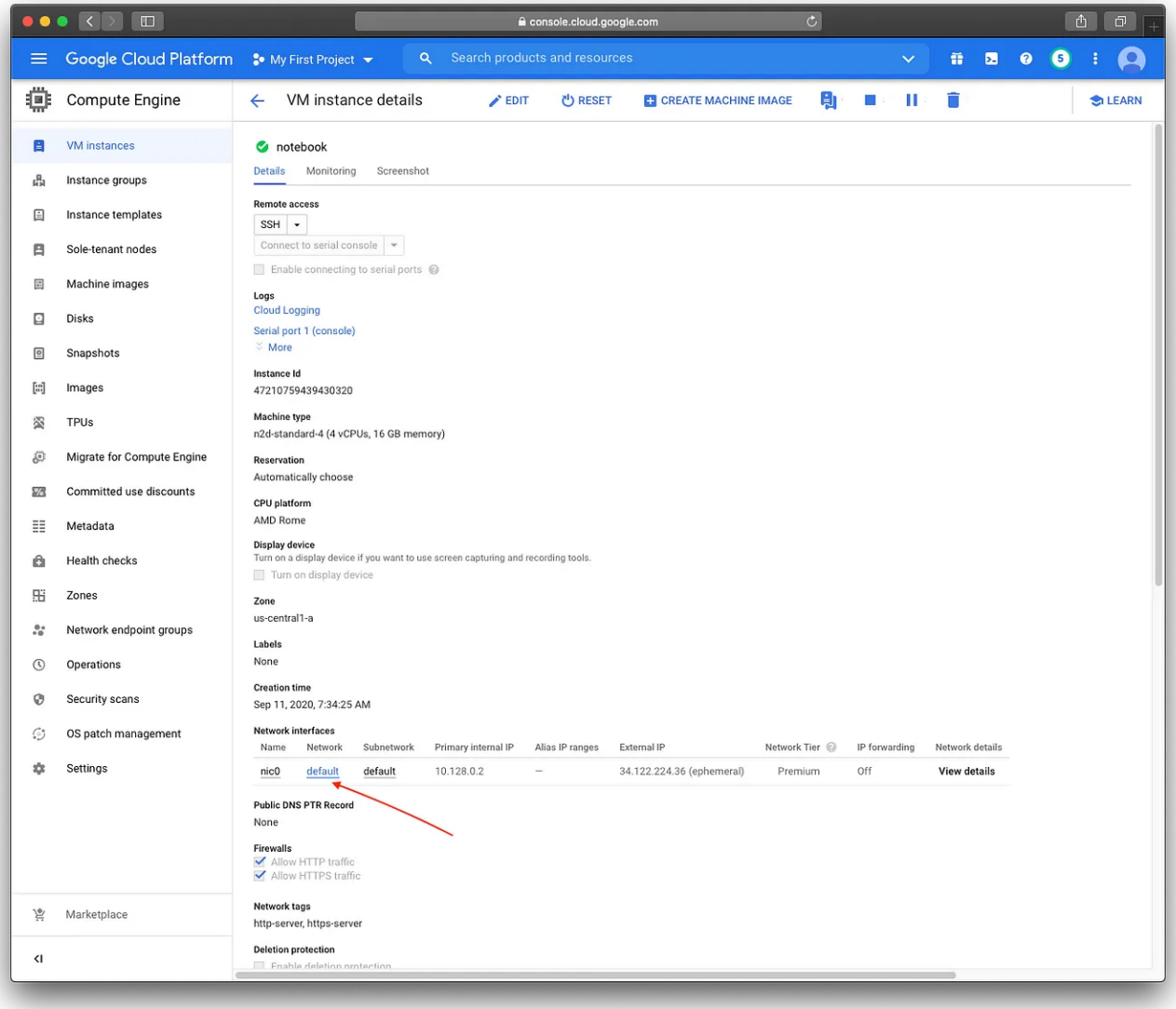
Step 2: Open port on firewall

I want to use port 60000 for this notebook so I will add a firewall rule

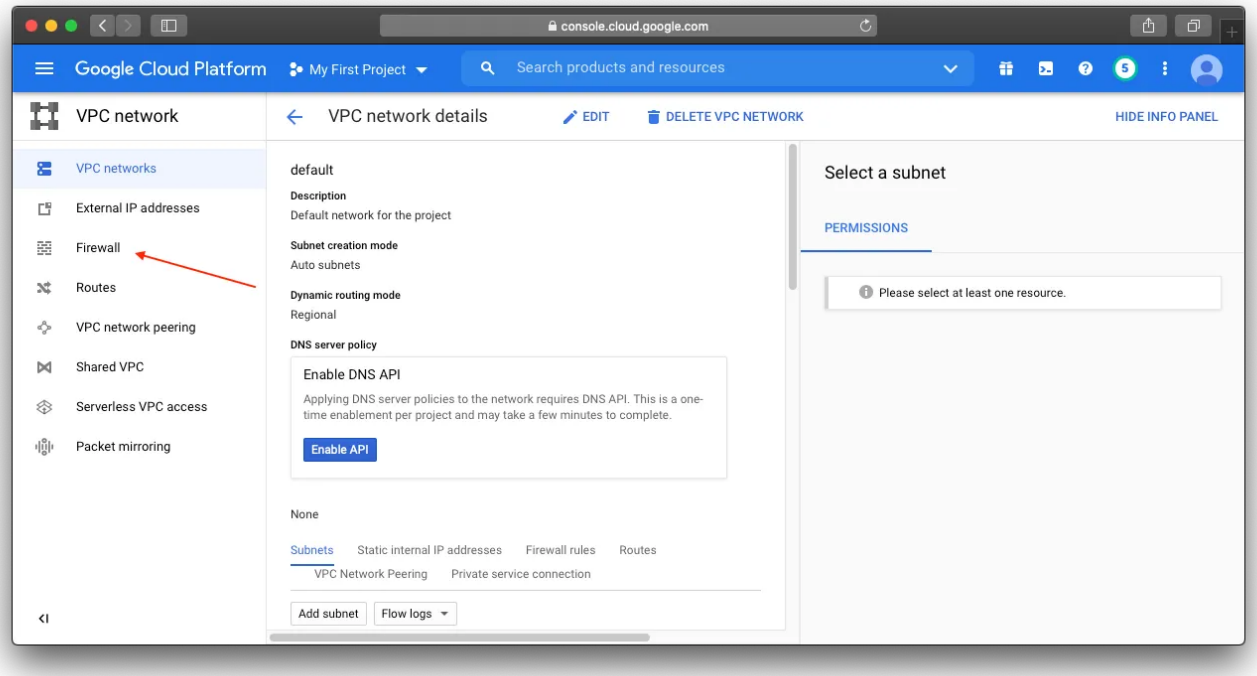
Click your instance



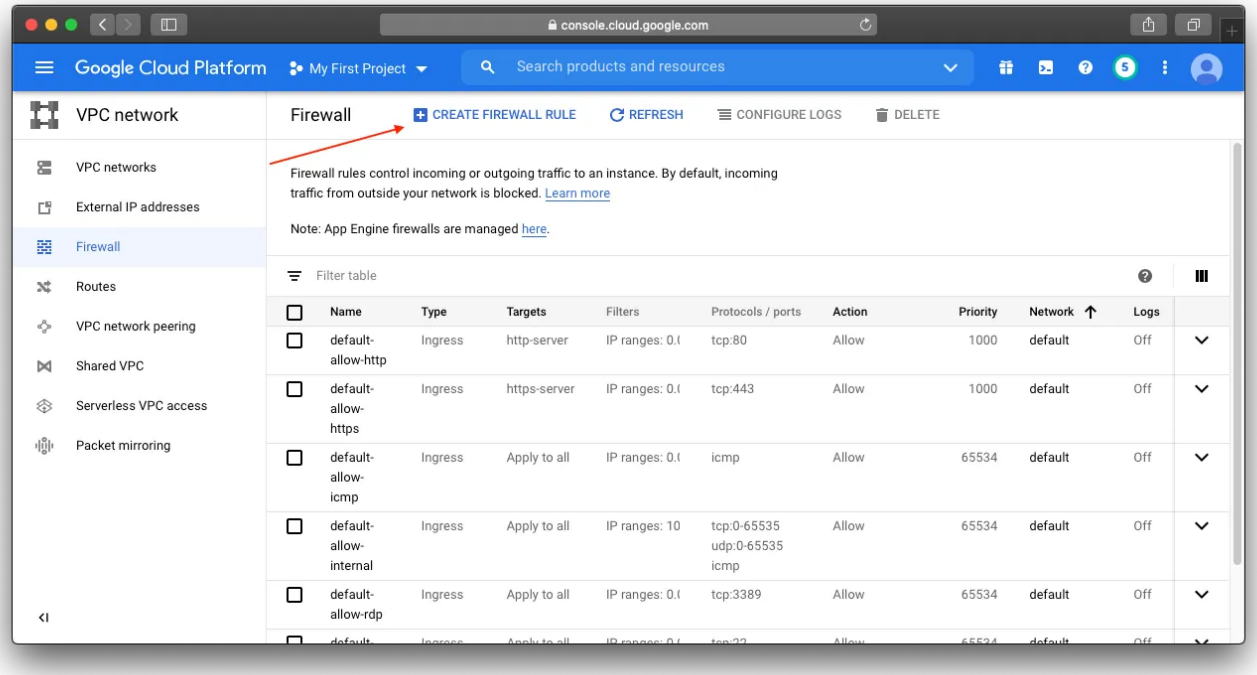
click network



Click Firewall

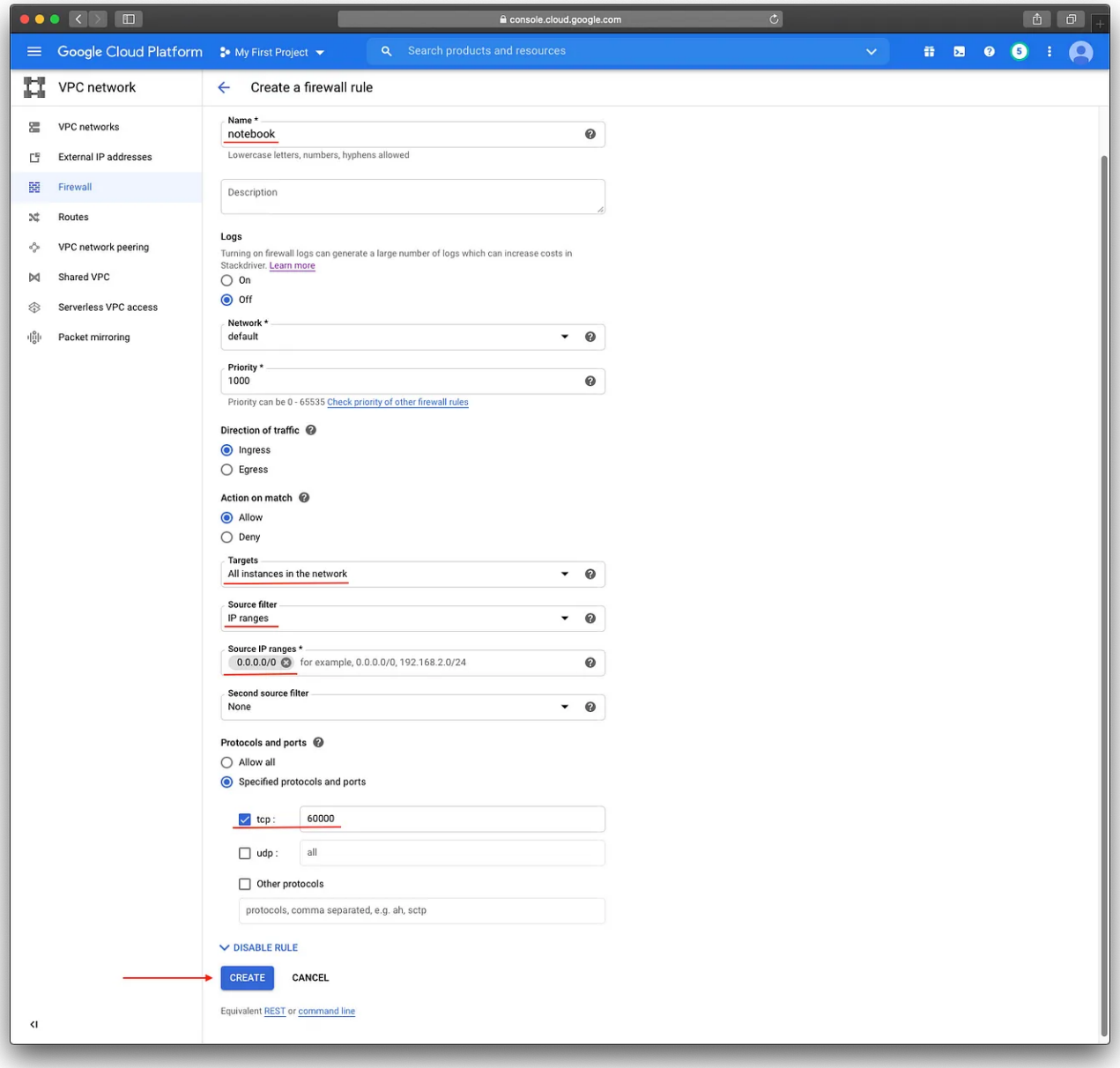


Click Create Firewall Rule



- give firewall rule a name. I write “notebook”
- targets: **All instances in the network**
- Source filter: **Ip ranges**
- Source IP ranges : **0.0.0.0/0**

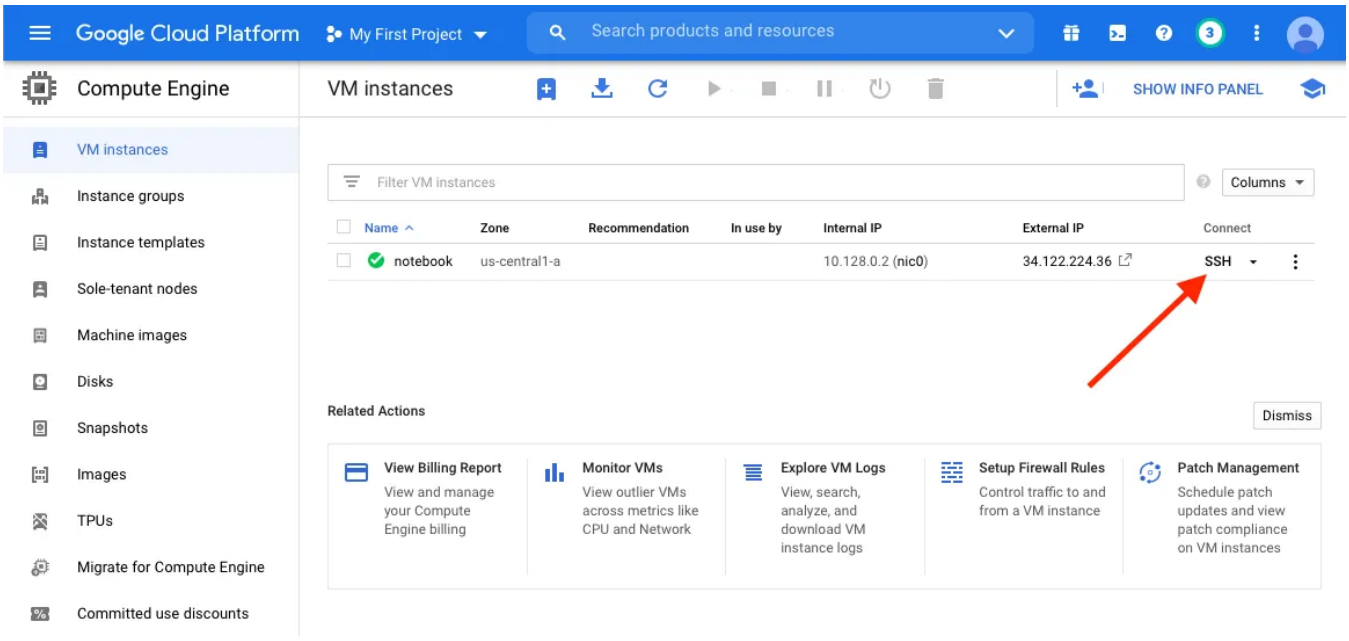
- second source filter : **none**
- Specified protocols and ports: **tcp:60000**



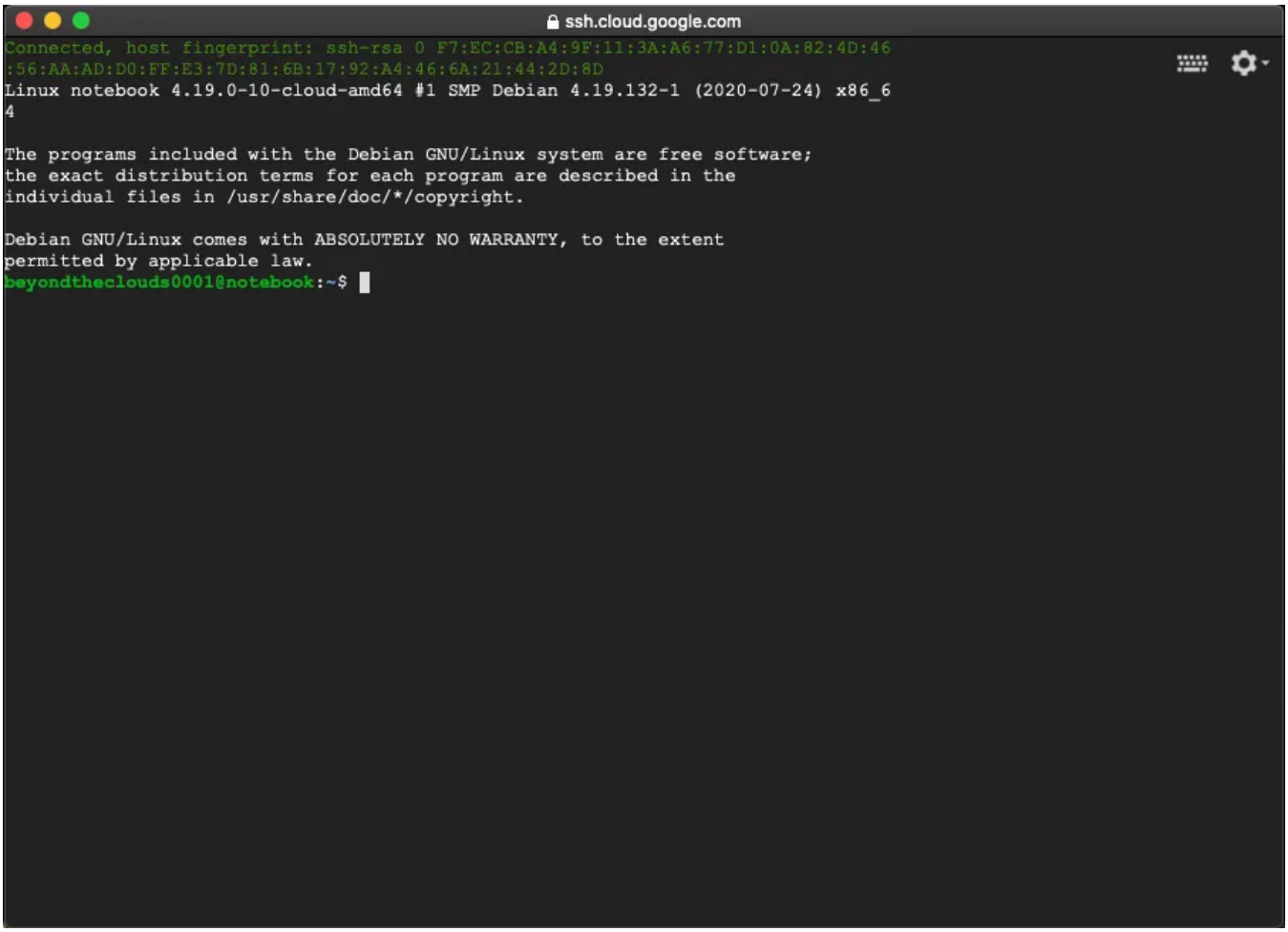
and click the **create** button. Now you can connect your notebook from tcp:60000.

Step 3: Install Jupyter Notebook

On the VM instances dashboard click the SSH button to connect your VM instance



now your browser will open another window which is your console



you need to enter all commands in this console let's start with an upgrade

```
sudo apt update && sudo apt upgrade -y
```

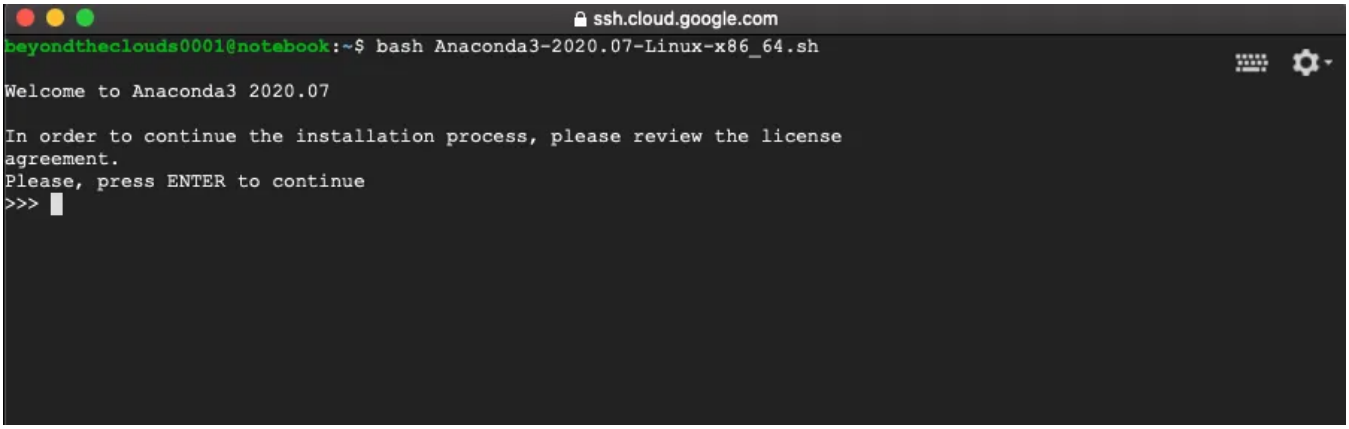
now visit this URL and find the latest release of Anaconda for Linux-x86_64 and right-click the link and copy it.

for now, Anaconda3–2021.11-Linux-x86_64.sh is the latest release. We use curl -O command to download the file

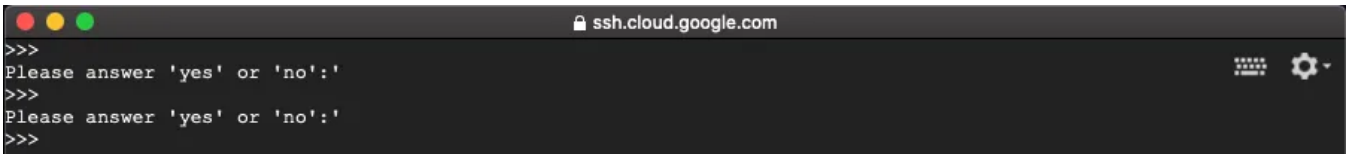
```
cd ~ && curl -O https://repo.anaconda.com/archive/Anaconda3-2021.11-Linux-x86_64.sh
```

this is a .sh file so we use the bash command to run it

```
bash Anaconda3-2021.11-Linux-x86_64.sh
```

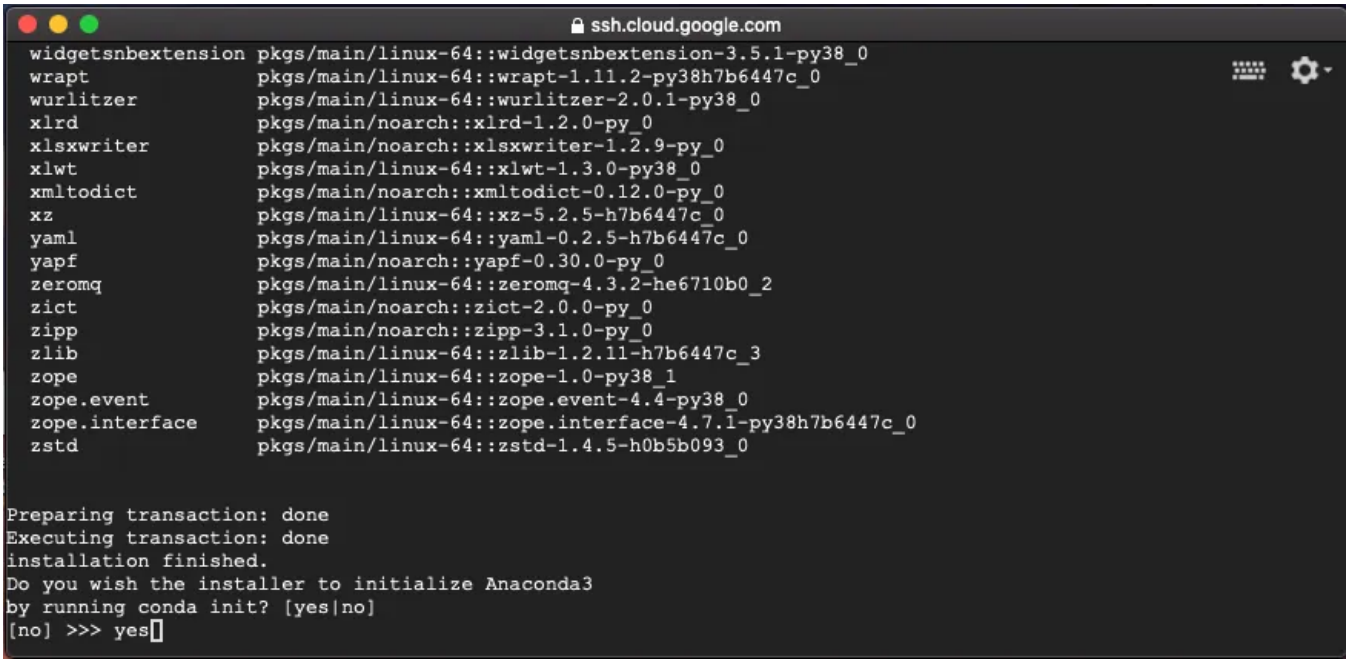


hold ENTER till you see yes or no choice and write **yes**



hit the ENTER for the last time to confirm installation location and your installation will start

After installation finished it will ask to run initialize Anaconda3 say **yes**

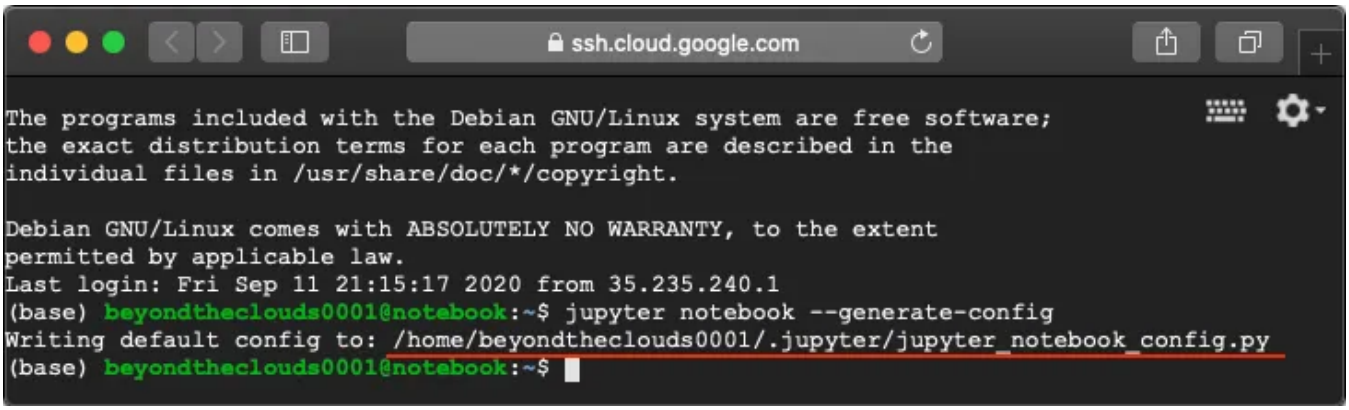


! important : close current shell and reconnect with ssh. Its because jupyter command not recognized with current ssh session.

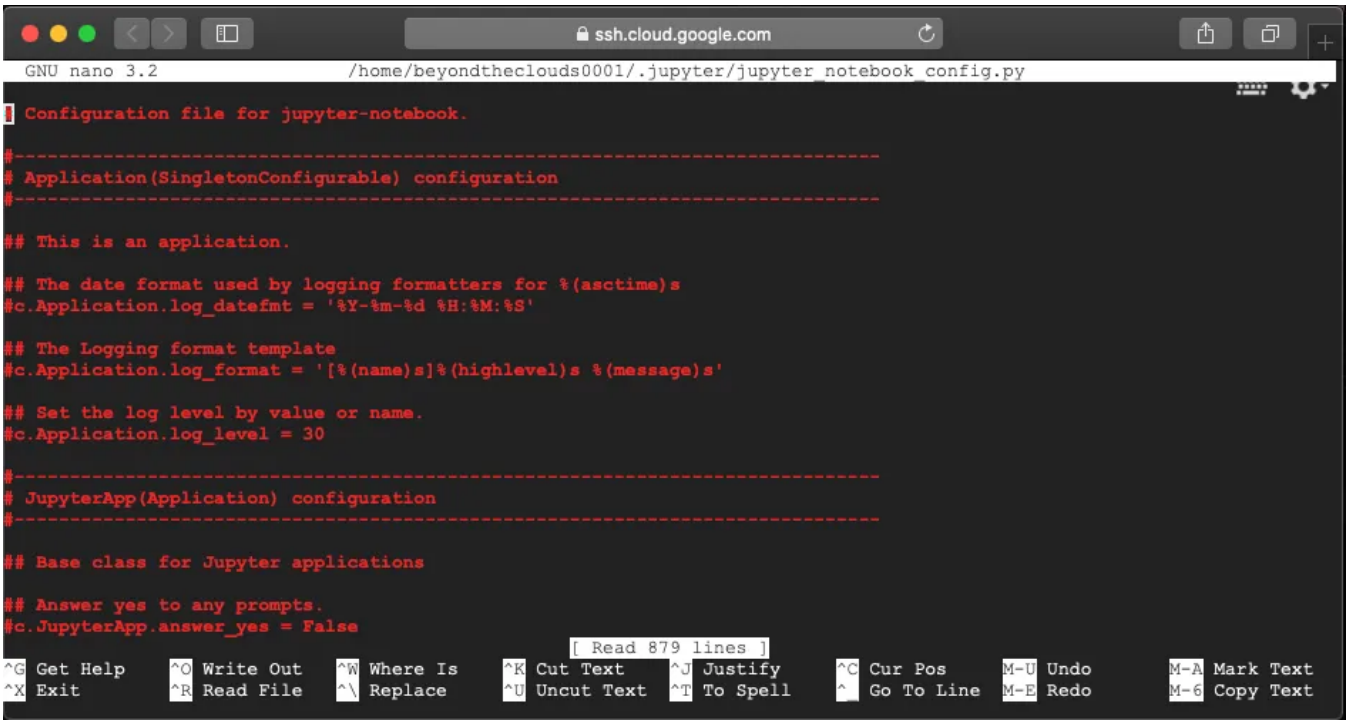
now we need to generate a jupyter notebook configuration file with this command

```
jupyter notebook --generate-config
```

After generating the config file command you will see the path of the jupyter_notebook_config.py file copy this path and edit it with nano file editor

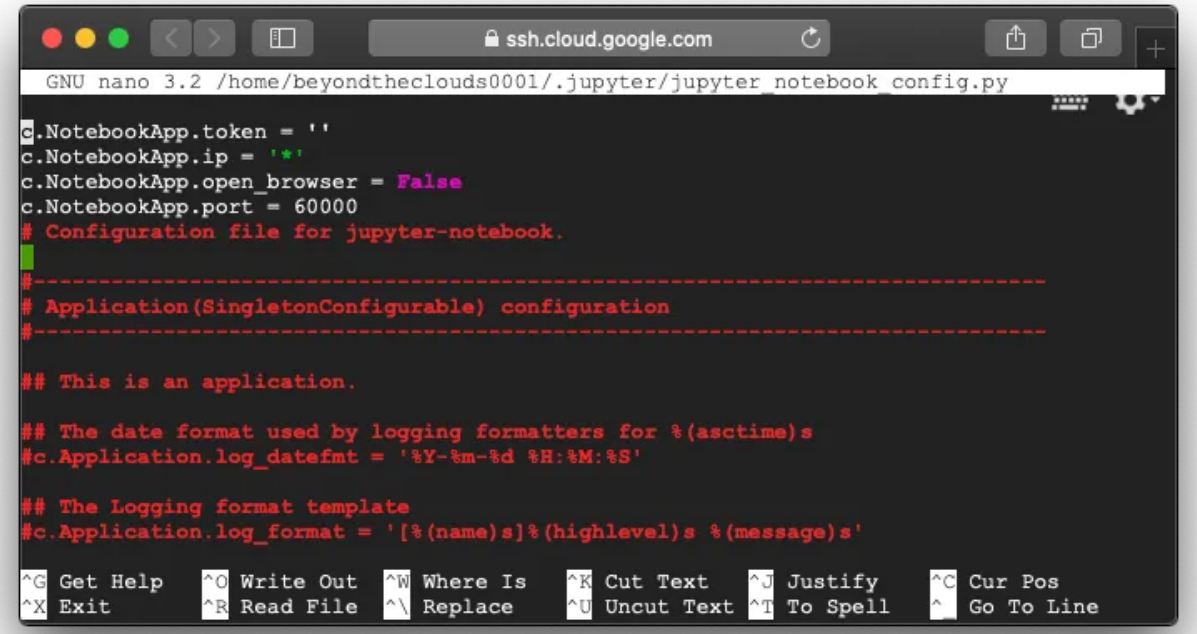



```
nano /home/beyondtheclouds0001/.jupyter/jupyter_notebook_config.py
```



you can simply copy and paste these lines at the beginning of the configuration file

```
c.NotebookApp.token = ''
c.NotebookApp.ip = '*'
c.NotebookApp.open_browser = False
c.NotebookApp.port = 60000
```

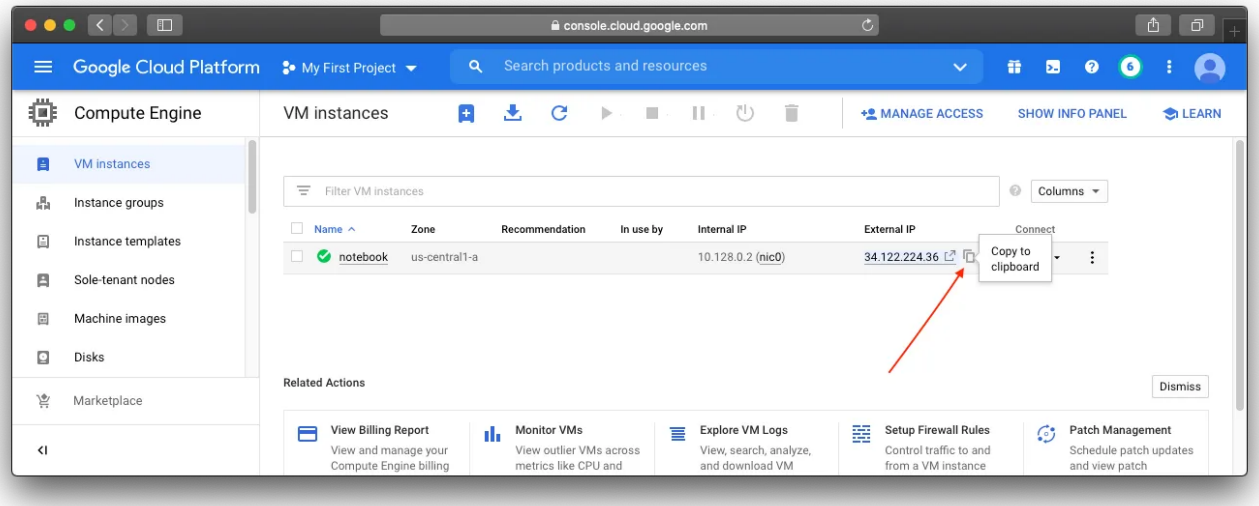


control + x to exit and press Y to save changes then press enter.

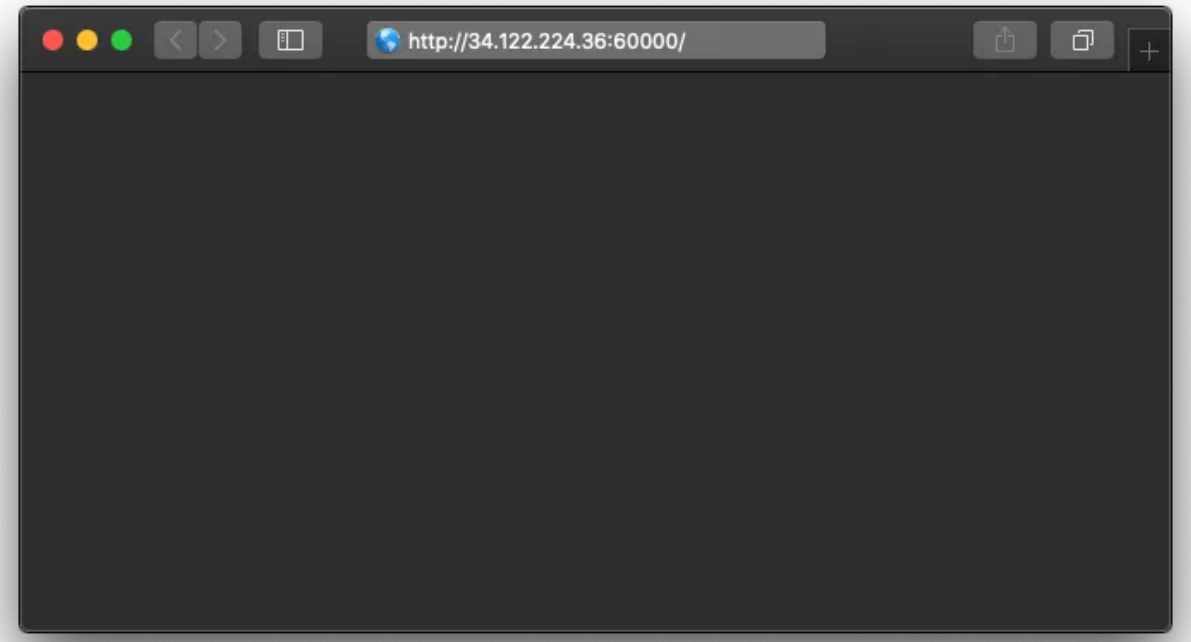
you are ready to go. Just start jupyter notebook with this command

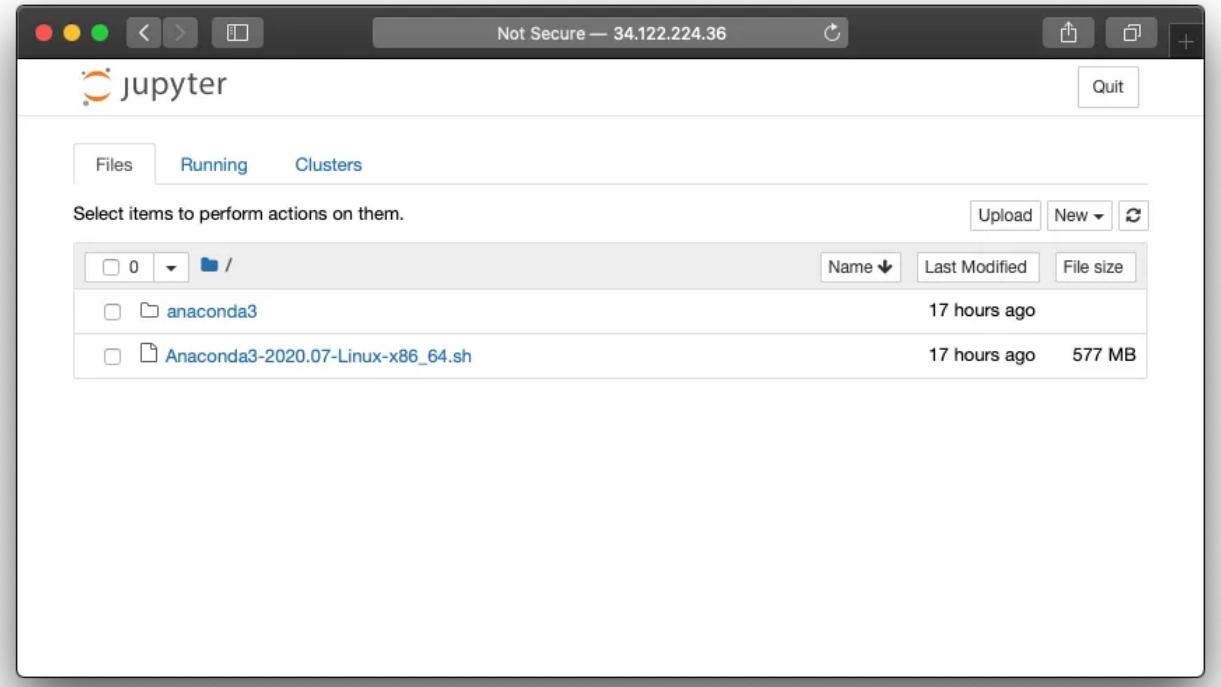
```
jupyter-notebook
```

Go to VM instances dashboard and copy your instance external IP address



open a new browser and enter your IP address and port number as it is





process constantly run on this virtual machine you need to convert this process to a Linux service.

Add a new file under the “/etc/systemd/system/” directory
this file name will be “jupyter.service”.

```
nano /etc/systemd/system/jupyter.service
```

change “**beyondtheclouds0001**” parts then copy-paste codes below to your notebook service file and save the file.

Note: in this example, my username on Linux is “beyondtheclouds0001”

where is my jupyter-notebook bin folder?
it is under the anaconda3 folder where you downloadAnaconda3–2021.11-Linux-x86_64.sh and run it. It auto-makes a folder in the same directory. In this example, I downloaded Anaconda to my home directory which is /home/beyondtheclouds0001/

```
[Unit]
Description=Jupyter Notebook

[Service]
Type=simple
PIDFile=/run/jupyter.pid
ExecStart=/home/beyondtheclouds0001/anaconda3/bin/jupyter-notebook --
config=/home/beyondtheclouds0001/.jupyter/jupyter_notebook_config.py
User=beyondtheclouds0001
Group=google-sudoers
WorkingDirectory=/home/beyondtheclouds0001/
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

so in the end, there will be a “/etc/systemd/system/jupyter.service” file

we need to reload daemon so linux system will see our new jupyter.service file

```
sudo systemctl daemon-reload
```

we need to make this service started when Linux makes a restart so run this command

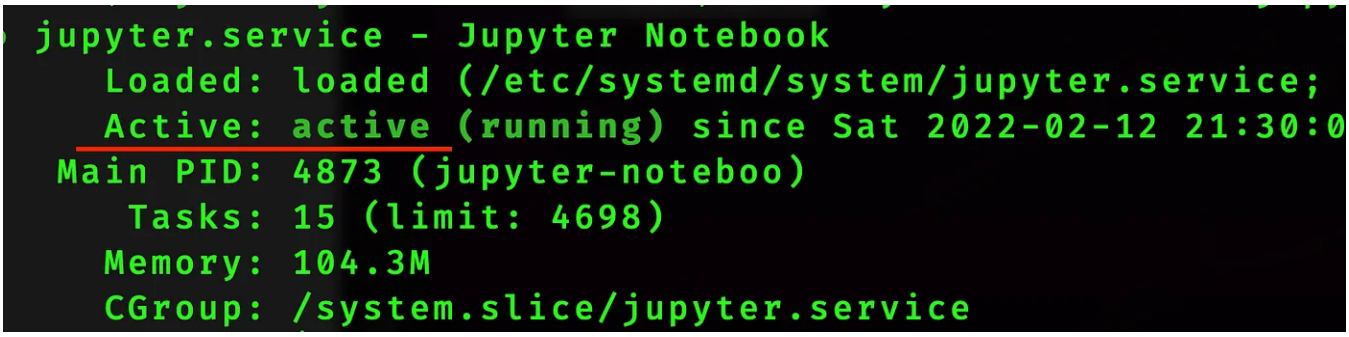
```
sudo systemctl enable jupyter
```

start jupyter service

```
sudo systemctl start jupyter
```

check if jupyter service is running

```
sudo systemctl status jupyter
```



these commands make your notebook service as an initial service and when you start your virtual machine notebook service will start automatically.

Do not hesitate to ask questions. keep coding 😜