

Assignments 1

Yog Chaudhary

11727095

ADTA 5750 Applied Natural Language Processing

Professor: Dr. Thuan L Nguyen

University of North Texas

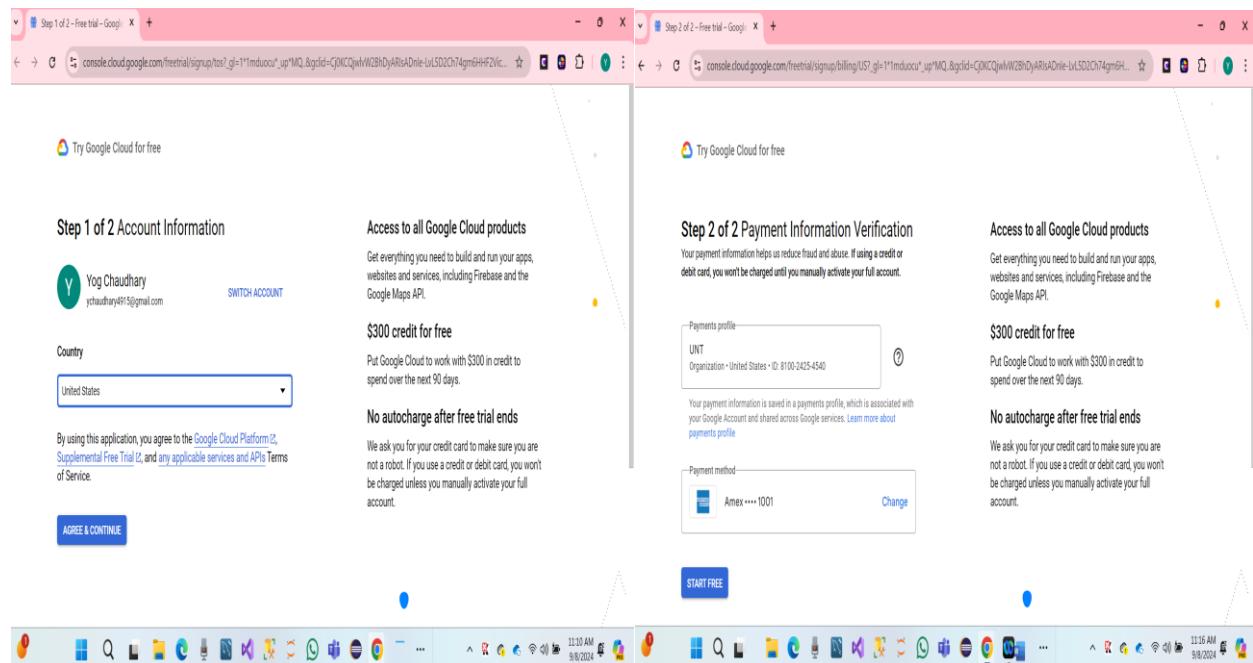
Sep 11, 2024

3. PART I: Select an Operating System (5 Points)

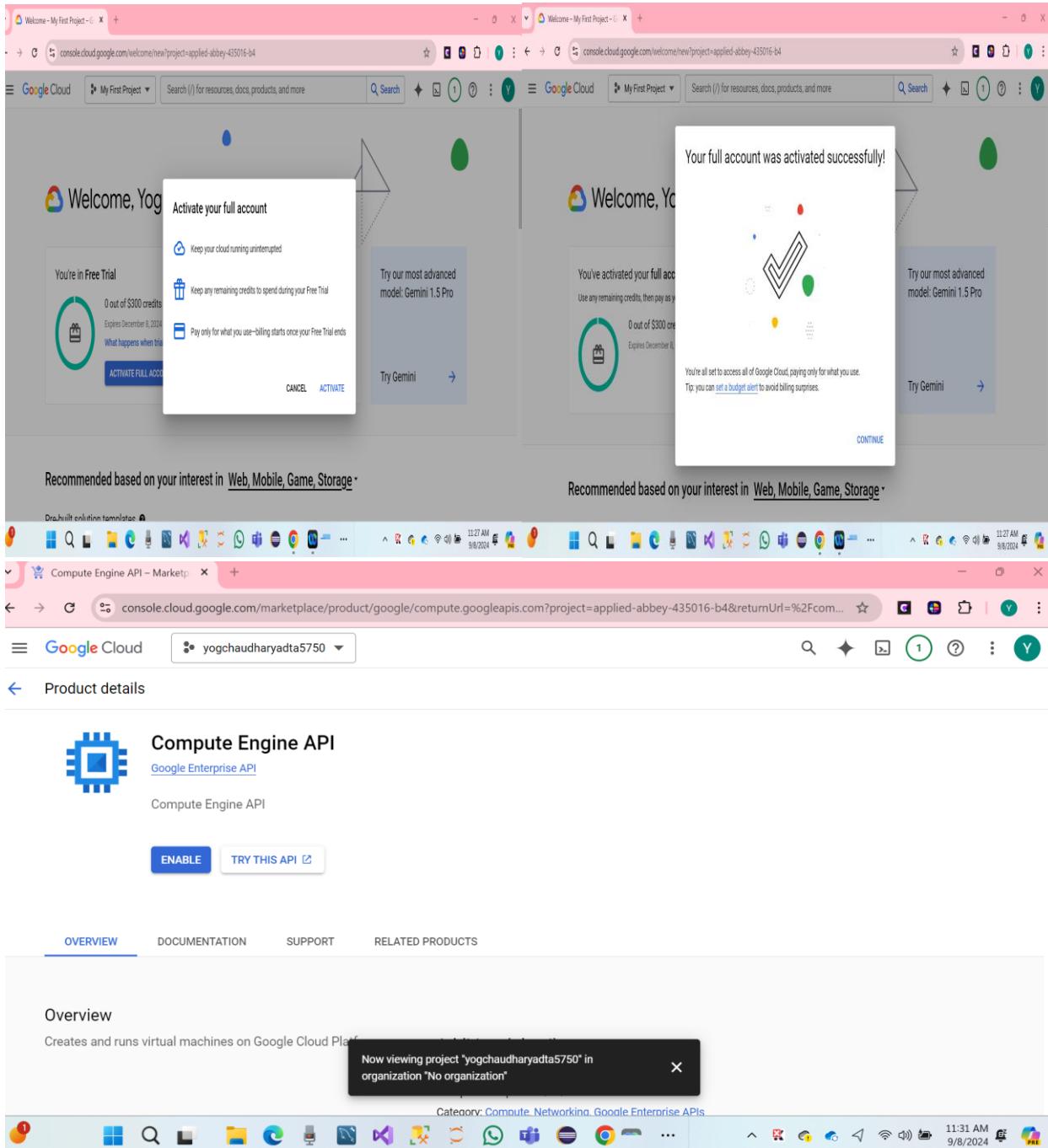
I will be using Windows as an OS for this course, and good knowledge of drive, directions and folders, files, the ownership status of these components, and administrative privileges. I can create new folders, access existing ones along with their actual contents, and create and save files within these folders. Download software applications, store them in specific folders, and install these applications in real, physical directories. Additionally, I can troubleshoot issues using various resources, such as online research, vendor support, and public technical forums. Therefore, I can use Windows effectively for cloud-based deep-learning tasks. Windows offers a user-friendly interface for all these actions, making it convenient for this course.

4. PART II: Set Up Deep Learning Virtual Machine (VM) in GCP (20 Points)

- Click on open GCP
- Click on the free Trail
- Enable computing engine.



- Click on Activate and continue.



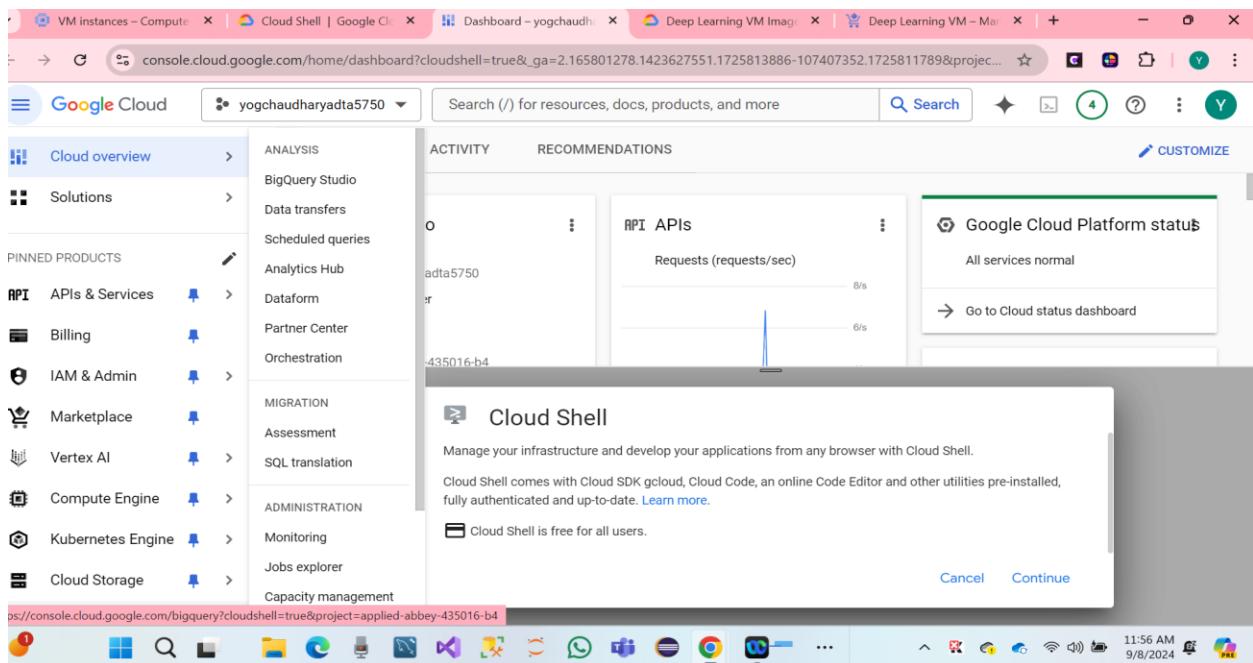
- Click on Enable computer engine.

The image shows two screenshots of the Google Cloud Compute Engine interface. The top screenshot displays the 'VM instances' page with a list of VM instances, including columns for Status, Name, Zone, Recommendations, In use by, Internal IP, External IP, and Connect. A large dashed cloud icon is centered on the page. The bottom screenshot shows a 'Select a project' dialog box over the main interface. The dialog has a search bar at the top and tabs for RECENT, STARRED, and ALL. It lists a single project: 'yogchaudharyadta5750' (ID: applied-abby-435016-b4). The main interface below the dialog shows the same 'VM instances' list as the top screenshot.

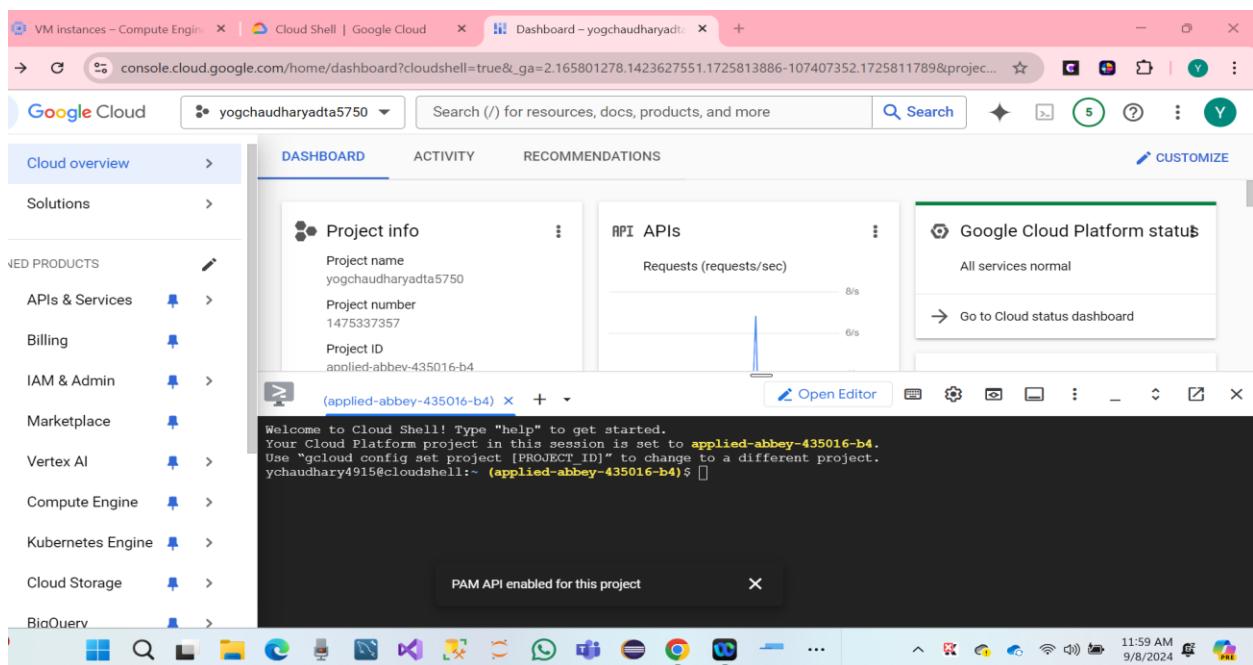
We have created **VM Instances**.

- Click on the search bar and click on cloud shell
- Click on Continue.
- Create Deep Learning VM Using GCP Deep Learning Image.
- Cloud Deep Learning VM Image.

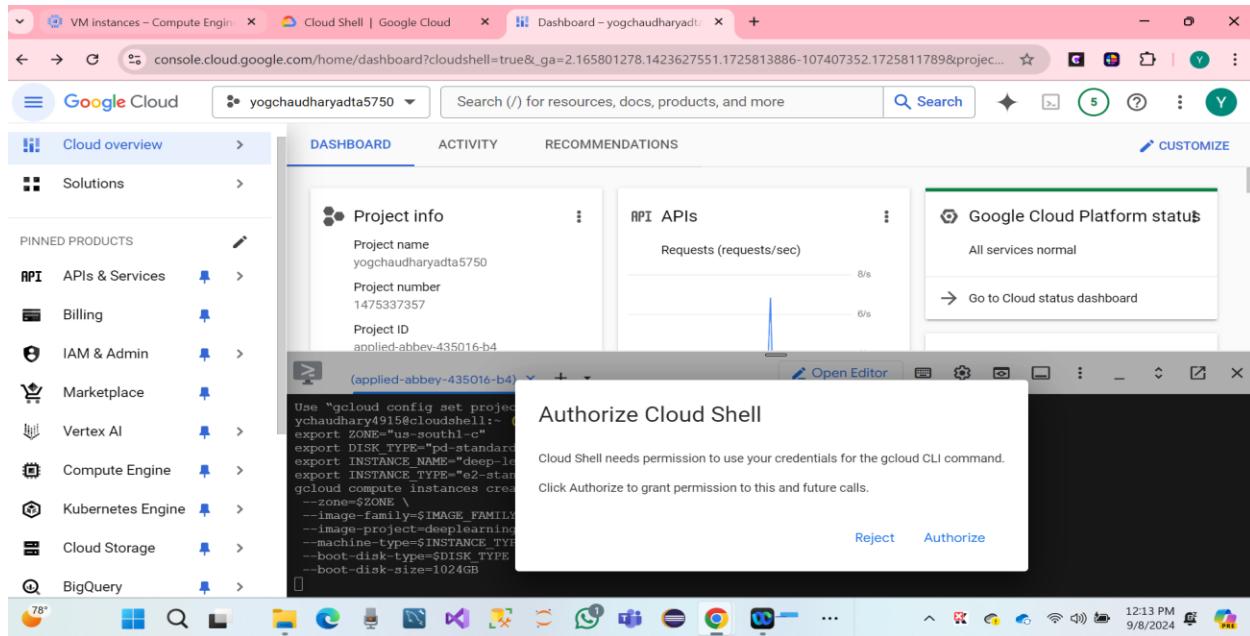
Click on link: <https://cloud.google.com/deep-learning-vm/>



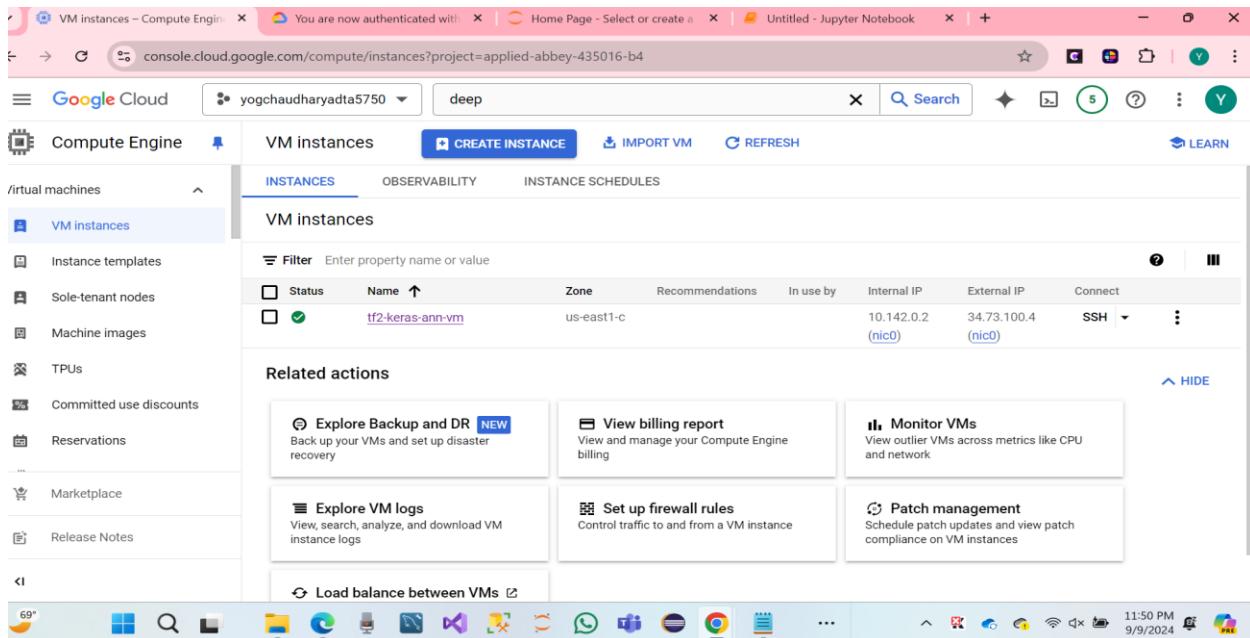
- Click on cloud shell to continue.



- We have a Drag tab to a new window.
- Open the cloud shell and copy and paste the code below.



- We have to click on Cloud Shell Authorized.
- Here we, our search bar showing Deep Learning console.
- Name of VM Instance **tf2-keras-ann-vm**
- Here Screenshot.



- Verify the TensorFlow Enterprises version of the VM Image.

The screenshot shows the Google Cloud Compute Engine interface. On the left, a sidebar lists options like 'Compute Engine', 'VM instances', 'Instance templates', etc. The main area displays the details for a VM named 'tf2-keras-ann-vm'. The 'Storage' tab is selected, showing the 'Boot disk' table:

Name	Image	Interface type	Size (GB)	Device name	Type	Architecture	Encryption
tf2-keras-ann-vm	tf-2-8-cpu-v20230925-debian-10-py37	SCSI	1024	tensorflow-vm-tmp1-boot-disk	Standard persistent disk	-	Google-managed

- Deep Learning VM on GCP has been set up successfully.

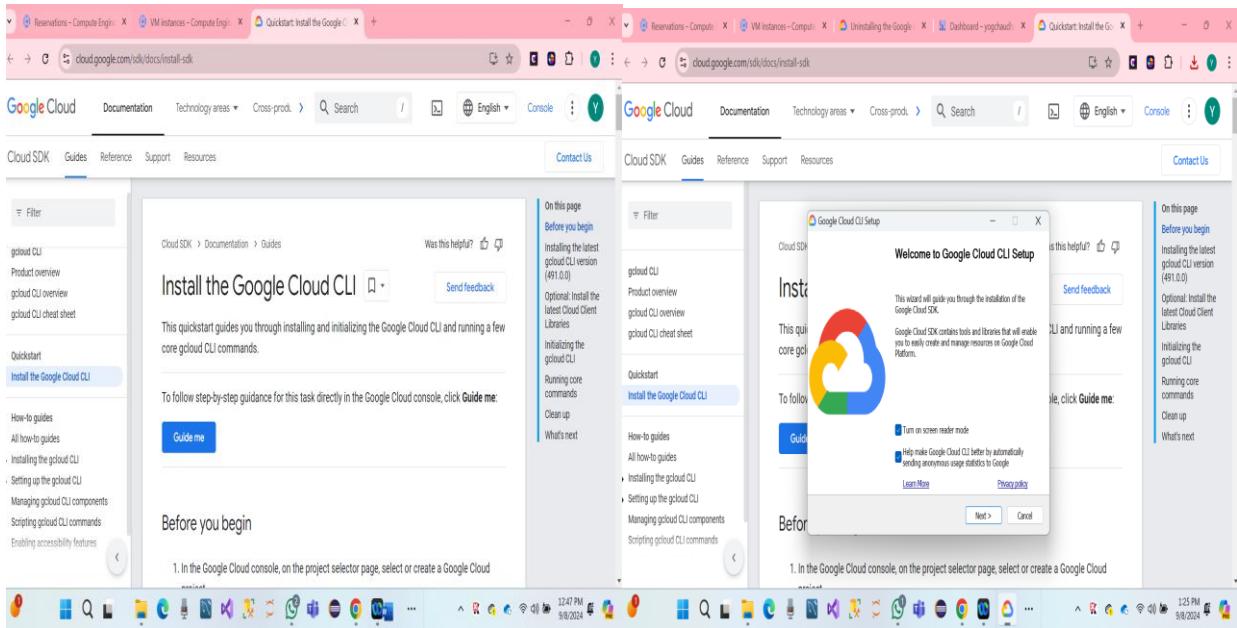
Access GCP Remote VM Console.

The screenshot shows the Google Cloud VM instances page. The sidebar includes 'Compute Engine', 'VM instances', 'Instance templates', etc. The main area lists the VM instance 'tf2-keras-ann-vm' in the 'us-east1-c' zone. A context menu is open over this instance, showing options like 'Start / Resume', 'Stop', 'Suspend', 'Reset', 'Delete', 'Create a group based on this VM', 'View network details', 'Create new machine image', 'View logs', and 'View monitoring'.

Note: Stop it when not Using.

Install GCLOUD SDK

- Click on the link: Download.

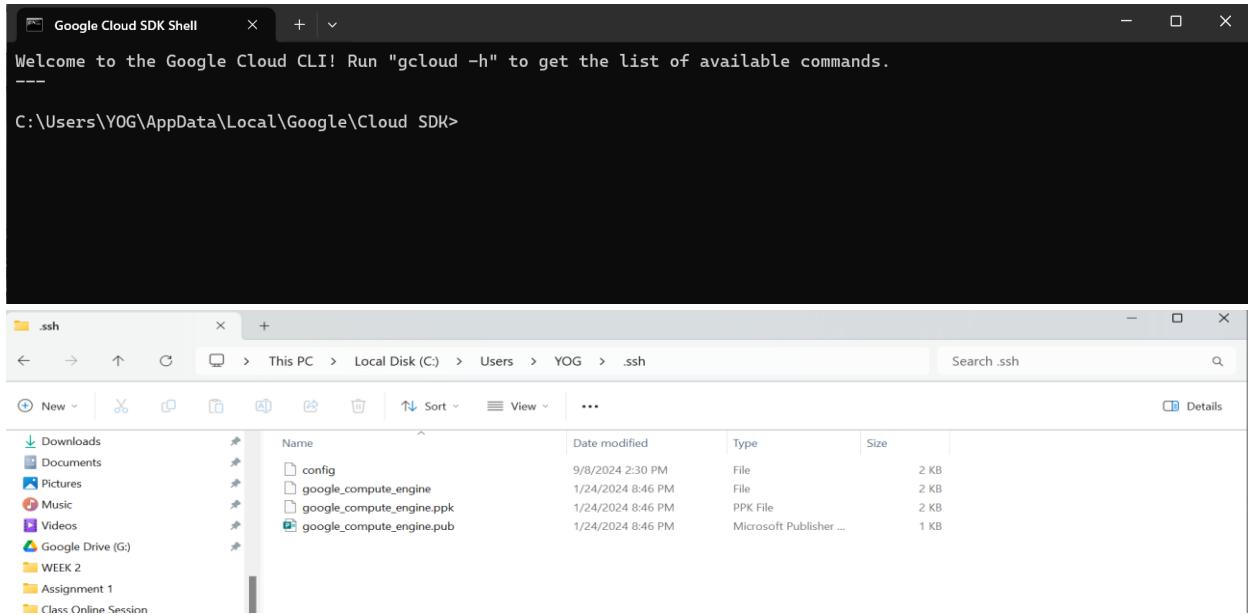


- We have to do a Finalize SDK Setting.

```
C:\WINDOWS\SYSTEM32\cmdc > + ^
You are signed in as: [ychaudhary4915@gmail.com].
Pick cloud project to use:
[1] applied-abbe-435016-b4
[2] Enter a project ID
[3] Create a new project
Please enter numeric choice or text value (must exactly match list item): 1
Your current project has been set to: [applied-abbe-435016-b4].
Do you want to configure a default Compute Region and Zone? (Y/n)? n
The Google Cloud CLI is configured and ready to use!
* Commands that require authentication will use ychaudhary4915@gmail.com by default
* Commands will reference project 'applied-abbe-435016-b4' by default
Run 'gcloud help config' to learn how to change individual settings
This gcloud configuration is called [default]. You can create additional configurations if you work with multiple accounts and/or projects.
Run 'gcloud topic configurations' to learn more.
Some things to try next:
* Run 'gcloud --help' to see the Cloud Platform services you can interact with. And run 'gcloud help COMMAND' to get help on any gcloud command.
* Run 'gcloud topic --help' to learn about advanced features of the CLI like arg files and output formatting
* Run 'gcloud cheat-sheet' to see a roster of go-to 'gcloud' commands.
C:\Users\YOG\AppData\Local\Google\Cloud SDK>
```

Finally, GCLOUD SDK Has been successful.

- Run GCLOUD SDK Shell
- Hare Screenshot.



5. PART III: Connect and Explore Remote VM Using SSH (5 Points)

Question 3.1:

Based on the lectures, open an SSH connection from the local computer to the remote VM

```
Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available commands.
---

C:\Users\YOG\AppData\Local\Google\Cloud SDK>gcloud auth login
Your browser has been opened to visit:

https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fccloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlservice.login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=8ue44iS5sSULx9Dpo1XXwBf097KPt&access_type=offline&code_challenge=LLVBEe3Wm_IABKsjCiTIui21-OgZkRSMrnG_vTwt2bQ&code_challenge_method=S256

You are now logged in as [lychaudhary4915@gmail.com].
Your current project is [applied-abbev-435016-b4]. You can change this setting by running:
$ gcloud config set project PROJECT_ID

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

```

Google Cloud SDK Shell
Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available commands.
---

C:\Users\YOG\AppData\Local\Google\Cloud SDK>gcloud auth login
Your browser has been opened to visit:

https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=ychaudhary4915@tf2-keras-ann-vm.~>
ct_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid
F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3At.
www.googleapis.com%2Fauth%2Fsqlservice.login+https%3A%2Faccounts.google.com%2Fauth%2Faccounts.reauth&state=rK9TiglpGt8s
Fjh7cAoT7AxpG84aYwgmCV0x9BQ&code_challenge_method=S
[1 04:42:11.187 NotebookApp] The port 8888 is already in use, trying another port
[1 04:42:11.188 NotebookApp] Serving notebooks from local directory: /home/ychaudhary4915
[1 04:42:11.188 NotebookApp] Jupyter Notebook 6.5.6 is running at:
[1 04:42:11.188 NotebookApp] http://localhost:8889/?token=598a8dbf44a2acfc6d7c75
[1 04:42:11.188 NotebookApp] or http://127.0.0.1:8889/?token=598a8dbf44a2acfc6d
[1 04:42:11.188 NotebookApp] Use Control-C to stop this server and shut down all
[1 04:42:11.188 NotebookApp] kernels (twice to skip confirmation).
[1 04:42:11.189 NotebookApp] No web browser found; could not locate runnable bro
wser.

You are now logged in as [ychaudhary4915@gmail.com]
Your current project is [applied-abby-435016-b4].
$ gcloud config set project PROJECT_ID
[1 04:42:11.193 NotebookApp]

C:\Users\YOG\AppData\Local\Google\Cloud SDK>gcloud
You should now be able to use ssh/scp with your inst
For example, try running:
$ ssh tf2-keras-ann-vm.us-east1-c.applied-abby-43
[1 04:42:11.193 NotebookApp]
To access the notebook, open this file in a browser:
file:///home/ychaudhary4915/.local/share/jupyter/runtime/nbserver-11149-
open.html
Or copy and paste one of these URLs:
http://localhost:8889/?token=598a8dbf44a2acfc6d7c753159884b5feef149fa48f
37850
or http://127.0.0.1:8889/?token=598a8dbf44a2acfc6d7c753159884b5feef149fa48f
37850

```

Question 3.2

Using the basic Linux command lines to explore the contents of the home directory

- Pwd
- whoami
- Ls -l
- Ls -all

```

ssh.cloud.google.com/v2/ssh/projects/applied-abby-435016-b4/zones/us-east1-c/instances/tf2-keras-ann-vm?authuser=0&hl=en_US&projectNumber=14753373
ssh.cloud.google.com/v2/ssh/projects/applied-abby-435016-b4/zones/us-east1-c/instances/tf2-keras-ann-vm?authuser=0&hl=en_US&projectNumber=14753373

SSH-in-browser

=====
Welcome to the Google Deep Learning VM
=====

Version: tf2-cpu.2-8.m112
Based on: Debian GNU/Linux 10 (buster) (GNU/Linux 4.19.0-25-cloud-amd64 x86_64)\n

Resources:
* Google Deep Learning Platform StackOverflow: https://stackoverflow.com/questions/tagged/google-dl-platform
* Google Cloud Documentation: https://cloud.google.com/deep-learning-vm
* Google Group: https://groups.google.com/forum/#!forum/google-dl-platform

To reinstall Nvidia driver (if needed) run:
sudo /opt/deeplearning/install-driver.sh
TensorFlow comes pre-installed with this image. To install TensorFlow binaries in a virtualenv (or conda env), please use the binaries that are pre-built for this image. You can find the binaries at /opt/deeplearning/binaries/tensorflow/
If you need to install a different version of Tensorflow manually, use the common Deep Learning image with the right version of CUDA

Linux tf2-keras-ann-vm 4.19.0-25-cloud-amd64 #1 SMP Debian 4.19.289-2 (2023-08-08) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
(base) ychaudhary4915@tf2-keras-ann-vm:~$ pwd
/home/ychaudhary4915
(base) ychaudhary4915@tf2-keras-ann-vm:~$ whoami
ychaudhary4915
(base) ychaudhary4915@tf2-keras-ann-vm:~$ ls -l
total 0
(base) ychaudhary4915@tf2-keras-ann-vm:~$ ls -all
total 36

```

Question 3.3

Create a new sub-folder named “JPTR_NTBK” under the home directory

- Mkdir JPTR_NTBK
- Cd JPTR_NTBK

```
ssh.cloud.google.com/v2/ssh/projects/applied-abby-435016-b4/zones/us-east1-c/instances/tf2-keras-ann-vm?authuser=0
ssh.cloud.google.com/v2/ssh/projects/applied-abby-435016-b4/zones/us-east1-c/instances/tf2-keras-a

SSH-in-browser

total 0
(base) ychaudhary4915@tf2-keras-ann-vm:~$ ls -all
total 36
drwxr-xr-x 6 ychaudhary4915 ychaudhary4915 4096 Sep 10 04:19 .
drwxr-xr-x 6 root root 4096 Sep 10 04:16 ..
-rw-r--r-- 1 ychaudhary4915 ychaudhary4915 220 Apr 18 2019 .bash_logout
-rw-r--r-- 1 ychaudhary4915 ychaudhary4915 3957 Sep 10 04:19 .bashrc
drwxr-xr-x 3 ychaudhary4915 ychaudhary4915 4096 Sep 10 04:19 .config
drwxr-xr-x 2 ychaudhary4915 ychaudhary4915 4096 Sep 10 04:19 .docker
drwx----- 3 ychaudhary4915 ychaudhary4915 4096 Sep 10 04:19 .gnupg
-rw-r--r-- 1 ychaudhary4915 ychaudhary4915 807 Apr 18 2019 .profile
drwx----- 2 ychaudhary4915 ychaudhary4915 4096 Sep 10 04:19 .ssh
(base) ychaudhary4915@tf2-keras-ann-vm:~$ mkdir JPTR_NTBK
(base) ychaudhary4915@tf2-keras-ann-vm:~$ cd JPTR_NTBK
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK$ ls
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK$ ls -all
total 8
drwxr-xr-x 2 ychaudhary4915 ychaudhary4915 4096 Sep 10 04:22 .
drwxr-xr-x 7 ychaudhary4915 ychaudhary4915 4096 Sep 10 04:22 ..
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK$ pwd
/home/ychaudhary4915/JPTR_NTBK
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK$ DATA
-bash: DATA: command not found
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK$ JP_NTBK
-bash: JP_NTBK: command not found
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK$ pwd
/home/ychaudhary4915/JPTR_NTBK
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK$ whoami
ychaudhary4915
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK$ ls -l
total 0
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK$ mkdir JP_NTBK
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK$ cd JP_NTBK
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK/JP_NTBK$ ls -l
total 0
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK/JP_NTBK$ DATA
```

Question 3.4

Change the current directory to the newly created folder

```
ssh.cloud.google.com/v2/ssh/projects/applied-abby-435016-b4/zones/us-east1-c/instances/tf2-keras-ann-vm?authuser=0&hl=en_US&projectNumber=1475337357&useAdminProxy=true - Google Chrome
ssh.cloud.google.com/v2/ssh/projects/applied-abby-435016-b4/zones/us-east1-c/instances/tf2-keras-ann-vm?authuser=0&hl=en_US&projectNumber=1475337357&useAdminProxy=true - Google Chrome

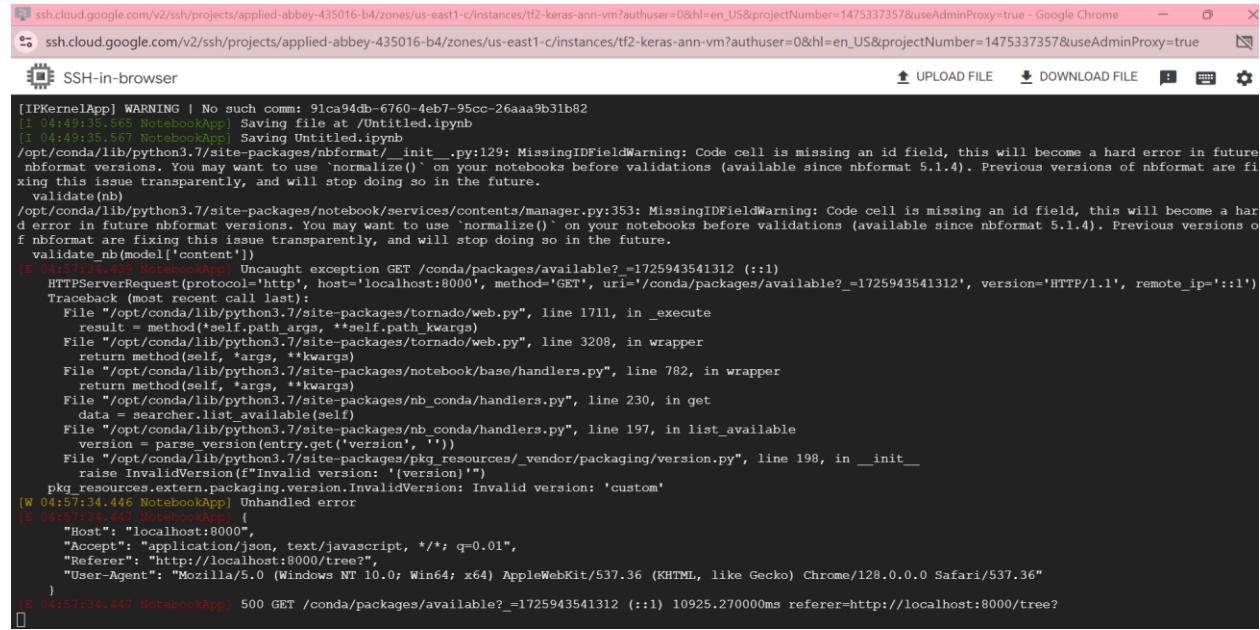
SSH-in-browser
UPLOAD FILE DOWNLOAD FILE

-bash: JP_NTBK: command not found
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK$ pwd
/home/ychaudhary4915/JPTR_NTBK
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK$ whoami
ychaudhary4915
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK$ ls -l
total 0
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK$ mkdir JP_NTBK
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK$ cd JP_NTBK
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK/JP_NTBK$ ls -l
total 0
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK/JP_NTBK$ DATA
-bash: DATA: command not found
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK/JP_NTBK$ JP_NTBK
-bash: JP_NTBK: command not found
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK/JP_NTBK$ la -all
-bash: la: command not found
(base) ychaudhary4915@tf2-keras-ann-vm:~/JPTR_NTBK/JP_NTBK$ jupyter notebook --port=8888
[I 04:33:04.861 NotebookApp] Writing notebook server cookie secret to /home/ychaudhary4915/.local/share/jupyter/runtime/notebook_cookie_secret
```

6. PART IV: Start and Connect to Jupyter Notebook in Remote VM (20 Points)

Question 4.1:

Based on the lectures, start the Jupyter Notebook server in the remote virtual machine

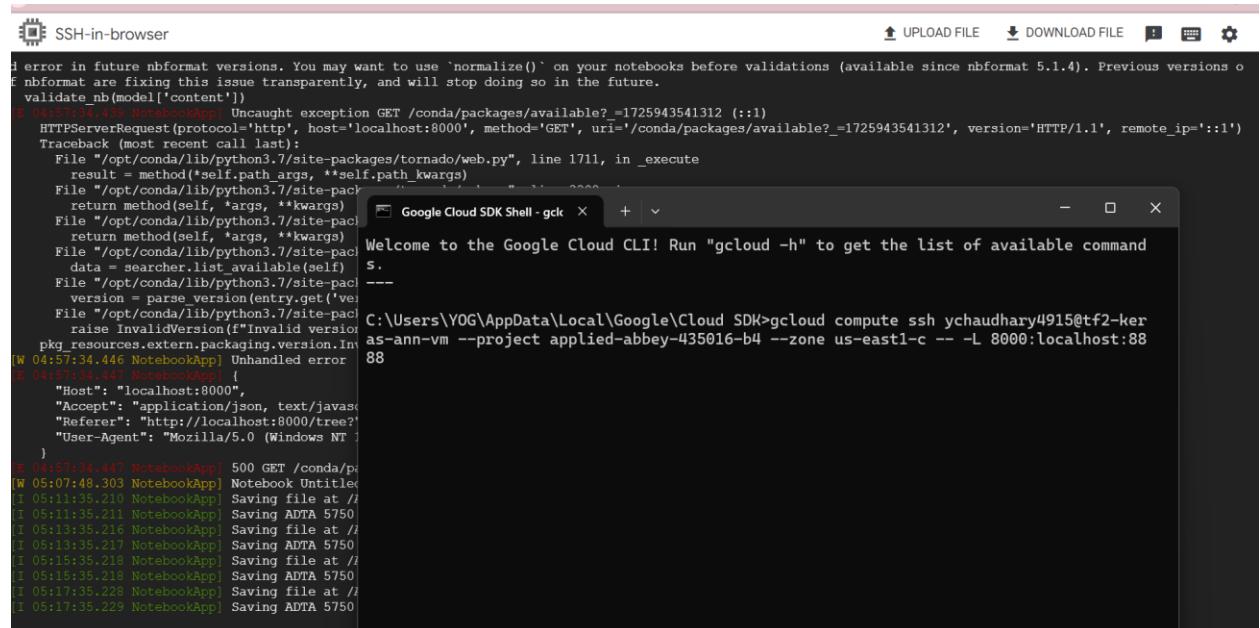


The screenshot shows a browser window with the URL `ssh.cloud.google.com/v2/ssh/projects/applied-abby-435016-b4/zones/us-east1-c/instances/tf2-keras-ann-vm?authuser=0&hl=en_US&projectNumber=1475337357&useAdminProxy=true`. The page displays Jupyter Notebook logs. The logs indicate that a user is trying to access the available packages endpoint at port 8000. The logs show several errors related to nbformat versions and missing ID fields. One error message is: "MissingIDFieldWarning: Code cell is missing an id field, this will become a hard error in future nbformat versions. You may want to use 'normalize()' on your notebooks before validations (available since nbformat 5.1.4). Previous versions of nbformat are fixing this issue transparently, and will stop doing so in the future." Another error message is: "HTTPServerRequest(protocol='http', host='localhost:8000', method='GET', url='/conda/packages/available?_=1725943541312', version='HTTP/1.1', remote_ip='::1')". The logs also mention "validate_nb(model['content'])" and "pkg_resources.extern.packaging.version.InvalidVersion('Invalid version: '(version)'")". The session ends with a 500 GET response.

```
[I 04:49:35.565 NotebookApp] Saving file at /Untitled.ipynb
[I 04:49:35.567 NotebookApp] Saving Untitled.ipynb
/opt/conda/lib/python3.7/site-packages/nbformat/_init_.py:129: MissingIDFieldWarning: Code cell is missing an id field, this will become a hard error in future nbformat versions. You may want to use 'normalize()' on your notebooks before validations (available since nbformat 5.1.4). Previous versions of nbformat are fixing this issue transparently, and will stop doing so in the future.
    validate(nb)
/opt/conda/lib/python3.7/site-packages/notebook/services/contentsmanager.py:353: MissingIDFieldWarning: Code cell is missing an id field, this will become a hard error in future nbformat versions. You may want to use 'normalize()' on your notebooks before validations (available since nbformat 5.1.4). Previous versions of nbformat are fixing this issue transparently, and will stop doing so in the future.
    validate_nb(model['content'])
[W 04:49:34.439 NotebookApp] Uncaught exception GET /conda/packages/available?_=1725943541312 (::1)
HTTPServerRequest(protocol='http', host='localhost:8000', method='GET', url='/conda/packages/available?_=1725943541312', version='HTTP/1.1', remote_ip='::1')
Traceback (most recent call last):
  File "/opt/conda/lib/python3.7/site-packages/tornado/web.py", line 1711, in _execute
    result = method(*self.path_args, **self.path_kwargs)
  File "/opt/conda/lib/python3.7/site-packages/tornado/web.py", line 3208, in wrapper
    return method(self, *args, **kwargs)
  File "/opt/conda/lib/python3.7/site-packages/notebook/base/handlers.py", line 782, in wrapper
    return method(self, *args, **kwargs)
  File "/opt/conda/lib/python3.7/site-packages/nb_conda/handlers.py", line 230, in get
    data = searcher.list_available(self)
  File "/opt/conda/lib/python3.7/site-packages/nb_conda/handlers.py", line 197, in list_available
    version = parse_version(entry.get('version', ''))
  File "/opt/conda/lib/python3.7/site-packages/pkg_resources/_vendor/packaging/version.py", line 198, in __init__
    raise InvalidVersion(f"Invalid version: '{version}'")
pkg_resources.extern.packaging.version.InvalidVersion: Invalid version: 'custom'
[W 04:57:34.446 NotebookApp] Unhandled error
[E 04:57:34.447 NotebookApp] {
  "Host": "localhost:8000",
  "Accept": "application/json, text/javascript, */*; q=0.01",
  "Referer": "http://localhost:8000/tree?",
  "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36"
}
[E 04:57:34.447 NotebookApp] 500 GET /conda/packages/available?_=1725943541312 (::1) 10925.270000ms referer=http://localhost:8000/tree?
□
```

Question 4.2:

`gcloud compute ssh ychaudhary4915@tf2-keras-ann-vm --project applied-abby-435016-b4 --zone us-east1-c -- -L 8000:localhost:8888`



The screenshot shows a terminal window with the command `gcloud compute ssh ychaudhary4915@tf2-keras-ann-vm --project applied-abby-435016-b4 --zone us-east1-c -- -L 8000:localhost:8888`. The terminal output shows a warning about nbformat versions and a stack trace for an unhandled error. The terminal then switches to a browser window titled "Google Cloud SDK Shell - gck" which displays the Google Cloud CLI welcome message: "Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available command". The browser also shows the Jupyter Notebook logs from the previous screenshot, indicating that the connection was successful and the port 8000 is listening.

```
[E 04:57:34.446 NotebookApp] Unhandled error
[E 04:57:34.447 NotebookApp] {
  "Host": "localhost:8000",
  "Accept": "application/json, text/javascript, */*; q=0.01",
  "Referer": "http://localhost:8000/tree?",
  "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36"
}
[E 04:57:34.447 NotebookApp] 500 GET /conda/packages/available?_=1725943541312 (::1) 10925.270000ms referer=http://localhost:8000/tree?
□
```

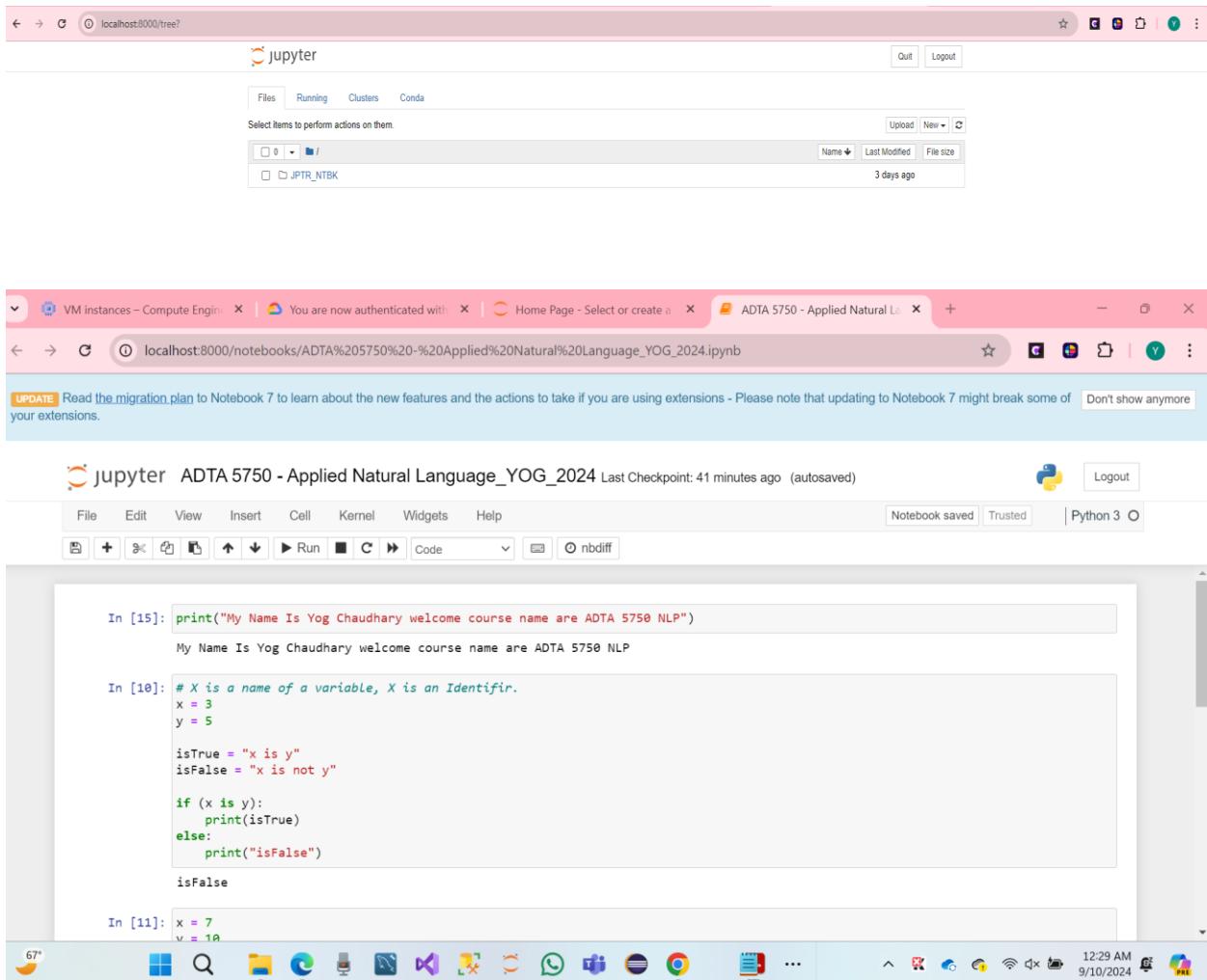
Google Cloud SDK Shell - gck

```
Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available command
S.
C:\Users\YOG\AppData\Local\Google\Cloud SDK>gcloud compute ssh ychaudhary4915@tf2-keras-ann-vm --project applied-abby-435016-b4 --zone us-east1-c -- -L 8000:localhost:8888
```

```
[W 04:57:34.446 NotebookApp] Unhandled error
[E 04:57:34.447 NotebookApp] {
  "Host": "localhost:8000",
  "Accept": "application/json, text/javascript, */*; q=0.01",
  "Referer": "http://localhost:8000/tree?",
  "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36"
}
[E 04:57:34.447 NotebookApp] 500 GET /conda/packages/available?_=1725943541312 (::1) 10925.270000ms referer=http://localhost:8000/tree?
□
```

Question 4.3:

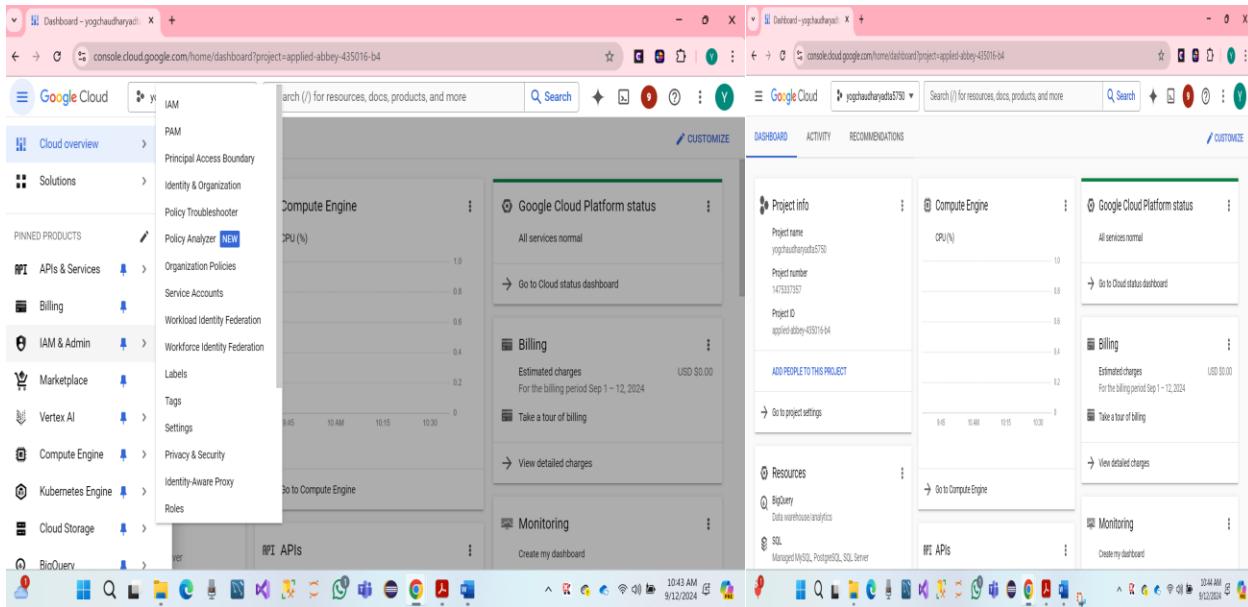
Use Jupyter Notebook that is currently running in the Remote Server (in a browser on the local computer).



7. PART V: Create Service Account for GCP Project (20 Points)

Create a Service Account for GCP, Project

- Click on Access Google Chrome browser and open the IAM & Admin Services accounts page.
- Click IAM & Admin to open a drop-down menu.
- Click on Select Services accounts.



- Click to select a project name **yogchaudharyadta5750**
- Here Screenshot.

Email	Status	Name	Description	Key ID	Key creation date	OAuth 2 Client ID	Actions
1475337357-compute@developer.gserviceaccount.com	Enabled	Compute Engine default service account	No keys	11074032	10:49 AM 9/12/2024		⋮

- Click to start a creative service account with an automotive name.

- We have a click to start a creative service account with an automotive name renamed.
- **service-acc-dl-tf-cnn**
- Click on Create and Services account details or permission to access the project role Owner.
- click on Create continues.

- Click on Selected Owner and Continue.

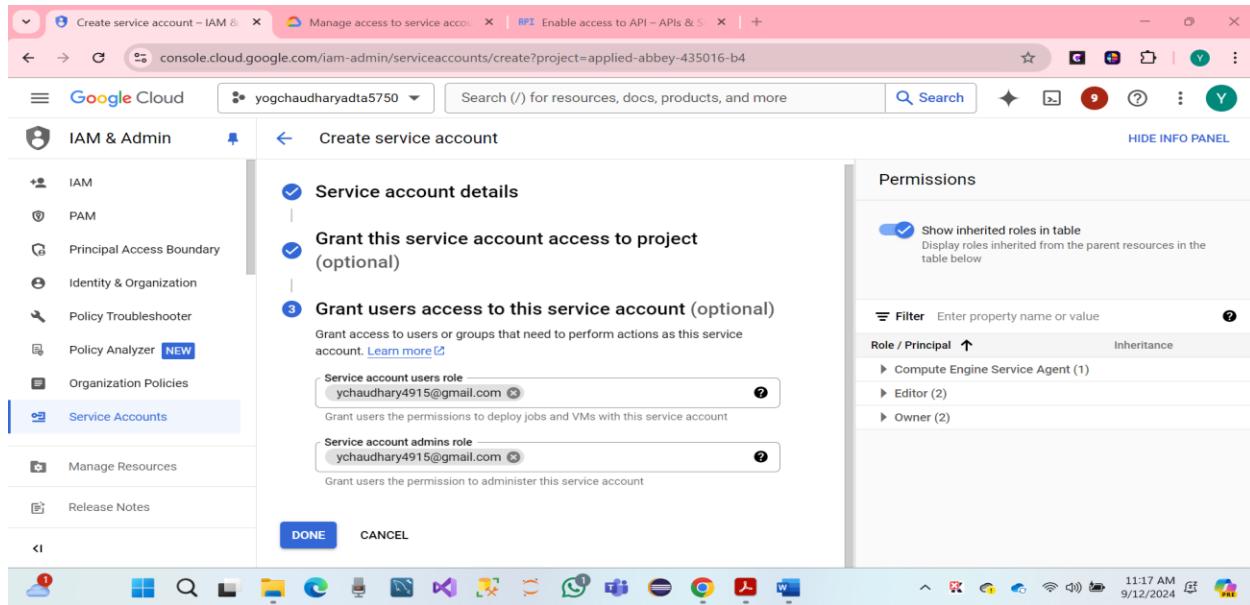
The screenshot shows the Google Cloud IAM & Admin interface. On the left sidebar, under the 'Service Accounts' section, the 'Create service account' button is highlighted. A modal window titled 'Grant users access to this service account (optional)' is open, prompting the user to 'Grant users access to this service account'. Below this, another modal window titled 'Service account created' is displayed. In the center of the screen, a dropdown menu titled 'Manage roles' is open, showing a list of roles: Compute Engine Service Agent, Editor, and Owner. The 'Owner' role is selected. The status bar at the bottom right shows the date as 9/12/2024 and the time as 11:10 AM.

The screenshot shows the Google Cloud IAM & Admin interface. The 'Service Accounts' section is active. A modal window titled 'Create service account' is open, showing the 'Service account details' step. It asks the user to 'Grant this service account access to project (optional)'. A dropdown menu for 'Role' is set to 'Owner'. Below it, a note states: 'Grant this service account access to yogchaudharyadta5750 so that it has permission to complete specific actions on the resources in your project.' A link to 'Learn more' is provided. The status bar at the bottom right shows the date as 9/12/2024 and the time as 11:12 AM.

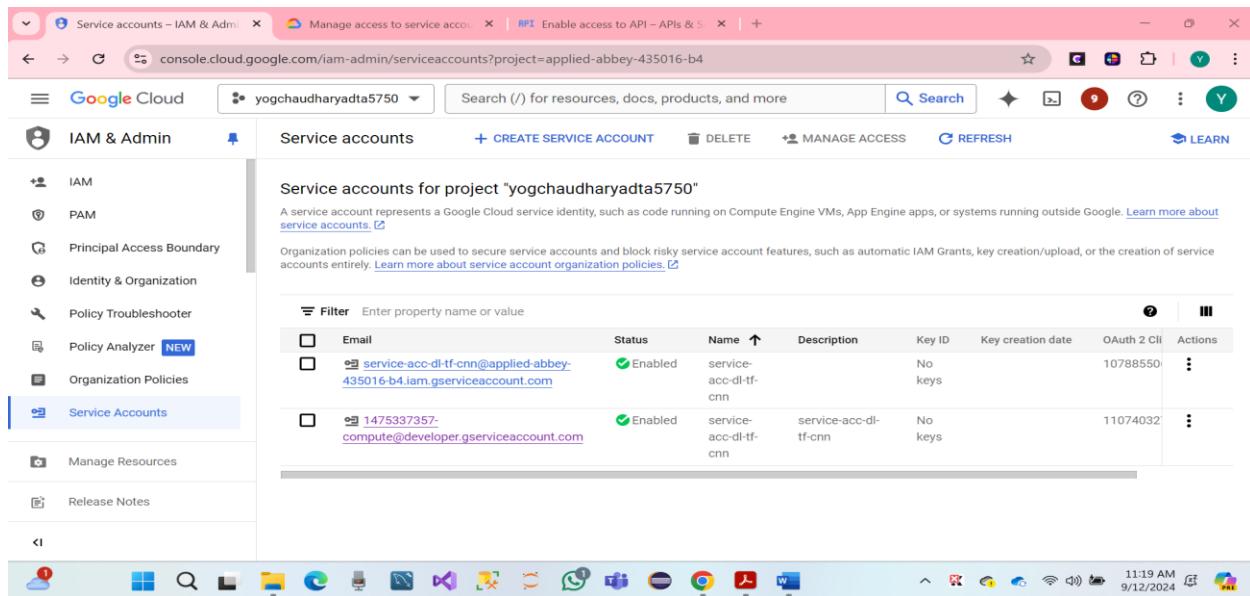
The screenshot shows the Google Cloud IAM & Admin service accounts page. The left sidebar is titled 'IAM & Admin' and includes sections for Service Accounts, Workload Identity Federation, Workforce Identity Federation, Labels, Tags, Settings, Privacy & Security, and Identity-Aware Proxy. Under 'Service Accounts', there are links for Manage Resources and Release Notes. The main content area is titled 'service-acc-dl-tf-cnn'. It has tabs for DETAILS, PERMISSIONS (which is selected), KEYS, METRICS, and LOGS. A note states: 'You can allow specific users to have ownership and access to service accounts and their settings. Users with the Owner or Editor basic role on the project can already modify service accounts, but you might want to restrict access for some users so that they can take only specific actions against service account resources.' Below this is a table titled 'VIEW BY PRINCIPALS' showing grants for the service account. The table has columns for Type, Principal, Name, Role, and Inheritance. It lists two entries: a Compute Engine Service Agent with the role 'Editor' and a user 'ychaudhary4915@gmail.com' with the role 'Owner'. There is also a 'VIEW BY ROLES' tab and a 'GRANT ACCESS' section with a 'REMOVE ACCESS' link. A filter bar allows entering a property name or value. The bottom right shows the date and time: 11:02 AM 9/12/2024.

The screenshot shows the 'Create service account' page. The left sidebar is titled 'IAM & Admin' and includes sections for IAM, PAM, Principal Access Boundary, Identity & Organization, Policy Troubleshooter, Policy Analyzer (NEW), Organization Policies, and Service Accounts (which is selected). The main content area is titled 'Create service account'. It has three steps: 1. Service account details (selected), 2. Grant this service account access to project (optional), and 3. Grant users access to this service account (optional). Step 1 is completed with a checkmark. Step 2 has a note: 'Grant access to users or groups that need to perform actions as this service account.' Step 3 has a note: 'Grant users the permission to administer this service account.' Below these steps are two input fields: 'Service account users role' (with a note: 'Grant users the permissions to deploy jobs and VMs with this service account') and 'Service account admins role' (with a note: 'Grant users the permission to administer this service account'). At the bottom are 'DONE' and 'CANCEL' buttons. To the right is a 'Permissions' panel with a checked checkbox for 'Show inherited roles in table' (which displays roles from the parent resources). A filter bar allows entering a property name or value. The bottom right shows the date and time: 11:15 AM 9/12/2024.

- Created to Grant user access to the service account name ychaudhary4915@gmail.com
- Hare Screenshot



- Click on Done and Continues.

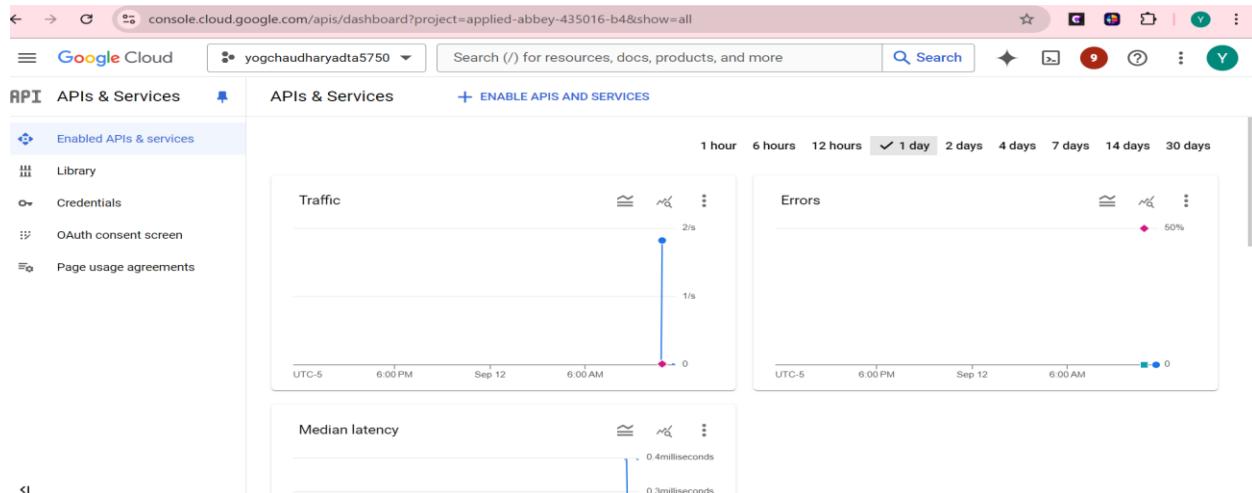


Finally, it's done, and the service account has been created successfully.

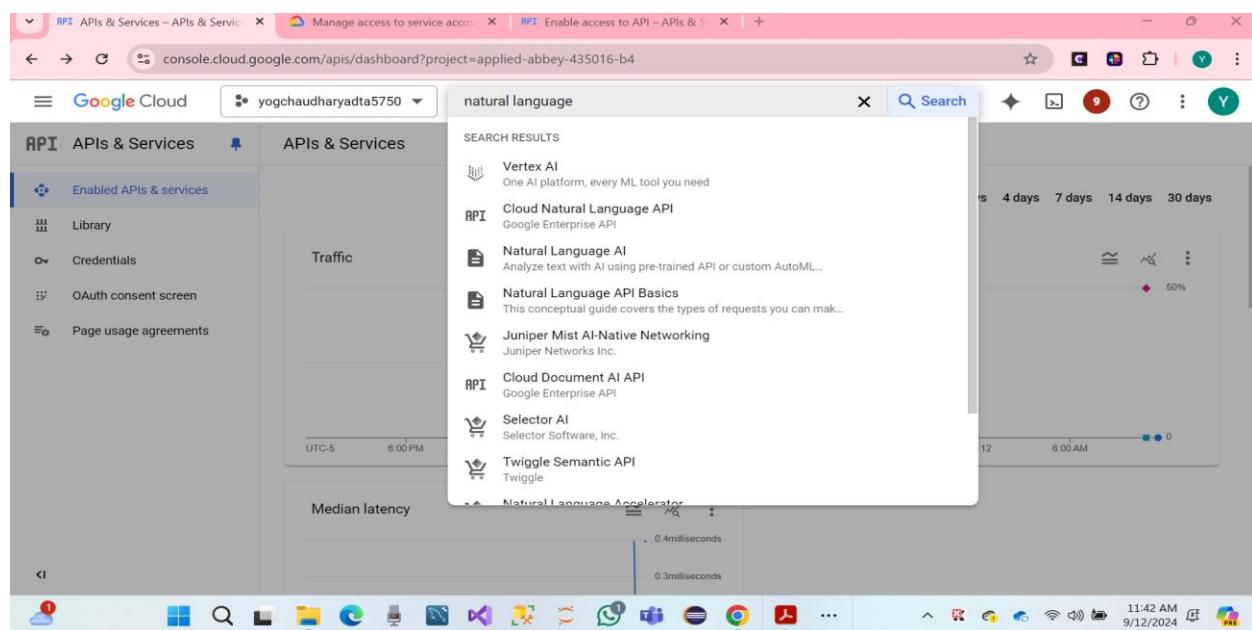
8. PART VI: Set Up Natural Language API for GCP Project (20 Points)

We have created a Natural Language API, Access GCP project Dashboard name project as **yogchaudharyadta5750**

- Click on Dashboard.
- Click on API Services.



- Click on API services Search the browser dropdown and enter Natural language.



- Again, we have to click on Enable for Cloud Natural Language API.

The screenshot shows the Google Cloud Marketplace page for the Cloud Natural Language API. At the top, there's a navigation bar with back, forward, and search icons. Below it, the title "Cloud Natural Language API" is displayed, along with its subtitle "Google Enterprise API". A brief description follows: "Provides natural language understanding technologies, such as sentiment analysis, entity..." Below the description are two buttons: "MANAGE" and "TRY THIS API". A green checkmark indicates "API Enabled". Underneath, there are tabs for "OVERVIEW" (which is selected), "PRICING", "DOCUMENTATION", and "RELATED PRODUCTS".

Overview

Provides natural language understanding technologies, such as sentiment analysis, entity recognition, entity sentiment analysis, and other text annotations, to developers.

Additional details

Type: [SaaS & APIs](#)
Last product update: 7/21/22

We have done a **Setup Services account for VM Instance**.

- Click on Name of VM Instance, and open and stop the remote deep learning server.

The screenshot shows the Google Cloud Compute Engine VM instances page. The left sidebar has a "Virtual machines" section with "VM instances" selected. The main area displays a table of VM instances. One instance, "tf2-keras-ann-vm", is listed with the following details:

Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
Green checkmark	tf2-keras-ann-vm	us-east1-c	C		10.142.0.2 (nic0)	35.185.16 (nic0)	SSH

Below the table, there's a "Related actions" section with several cards:

- Explore Backup and DR (NEW)
- View billing report
- Monitor VMs
- Explore VM logs
- Set up firewall rules
- Patch management

The screenshot shows the Google Cloud Compute Engine VM instances page. The left sidebar is titled 'Virtual machines' and includes options like 'VM instances', 'Instance templates', 'Sole-tenant nodes', 'Machine images', 'TPUs', 'Committed use discounts', 'Reservations', and 'Marketplace'. The main area has tabs for 'INSTANCES', 'OBSERVABILITY', and 'INSTANCE SCHEDULES'. Under 'INSTANCES', there is a table with one row for 'tf2-keras-ann-vm'. The table columns include Status (Status), Name (Name ↑), Zone (Zone), Recommendations (Recommendations), In use by (In use by), Internal IP (Internal IP), External IP (External IP), and Connect (Connect). The instance details are: Status (Running), Name (tf2-keras-ann-vm), Zone (us-east1-c), Internal IP (10.142.0.2 (nic0)), External IP (SSH), and Connect (nic0). Below the table, there is a section titled 'Related actions' with links to 'Explore Backup and DR', 'View billing report', and 'Monitor VMs'.

- We created the information page of the VM instances is Opened and Stopped.

The screenshot shows the Google Cloud Compute Engine VM instance details page for 'tf2-keras-ann-vm'. The left sidebar is identical to the previous screenshot. The main area has tabs for 'DETAILS', 'OBSERVABILITY', 'OS INFO', and 'SCREENSHOT'. The 'DETAILS' tab is selected. It shows basic information about the instance, including its name, instance ID, description, type, status, creation time, zone, instance template, in use by, and reservations. The status is listed as 'Stopped'. The 'Logs' section shows a log entry for 'Serial port 1 (console)'. The 'Basic information' section provides detailed settings for the instance.

- Click to open edit on the top menu bar.

The screenshot shows the 'Edit tf2-keras-ann-vm instance' page in the Google Cloud Compute Engine interface. The left sidebar lists 'Virtual machines' options like VM instances, Instance templates, Sole-tenant nodes, Machine images, TPUs, Committed use discounts, Reservations, Marketplace, and Release Notes. The main panel displays 'Basic information' for the instance, including its ID (6088690326467410180), status (Stopped), creation time (Sep 9, 2024, 11:15:40 PM UTC-05:00), zone (us-east1-c), reservation (Automatically choose (default)), and confidential VM service (Disabled). It also includes a 'Rename' section where the VM instance name is set to 'tf2-keras-ann-vm'. A note below says 'Tip: Reference the VM by its URI in API calls and gcloud CLI commands to make sure your project isn't affected by any name changes.' Below this is a 'Remote access' section with a checkbox for 'Enable connecting to serial ports'. At the bottom are 'SAVE' and 'CANCEL' buttons.

Get credentials for the service account Open the IAM & Admin Service account page and open a drop-down menu.

- Click on the open service account.

The screenshot shows the 'Service accounts - IAM & Admin' page in the Google Cloud IAM & Admin interface. The left sidebar lists 'IAM & Admin' options like Policy Analyzer, Organization Policies, Service Accounts (which is selected), Workload Identity Federation, Workforce Identity Federation, Labels, Tags, Settings, Manage Resources, and Release Notes. The main panel displays 'Service accounts for project "yogchaudharyadta5750"'. It shows two service accounts: 'service-acc-dl-tf-cnn@applied-abbe...435016-b4.iam.gserviceaccount.com' (Enabled, no keys) and '1475337357-compute@developer.iam.gserviceaccount.com' (Enabled, no keys). A 'VM instance started' notification is visible at the bottom. At the top right, there are buttons for '+ CREATE SERVICE ACCOUNT', 'DELETE', 'MANAGE ACCESS', 'REFRESH', and 'LEARN'.

- Copy email: service-acc-dl-tf-cnn@applied-abbe...435016-b4.iam.gserviceaccount.com

The screenshot shows the Google Cloud IAM & Admin interface for a service account named 'service-acc-dl-tf-cnn'. The 'Service account details' section is visible, showing the account's name, email (service-acc-dl-tf-cnn@applied-abby-435016-b4.iam.gserviceaccount.com), unique ID (107885506064580655523), and status (Enabled). The 'Service account status' section indicates that disabling the account preserves policies without deletion.

We have a create Key file JSON of Service accounts.

- Open an SSH connection to the remote deep-learning server
- Run Command below line
- Ls -all

The screenshot shows the Google Cloud Compute Engine VM instances page. A terminal window is open, displaying the output of the 'ls -all' command. The output includes several files, notably 'key.json' which is explicitly mentioned in the list.

```

-rw-r--r-- 1 ychauhdary4915 ychauhdary4915 3957 Sep 10 04:19 .bashrc
drwxr-xr-x 3 ychauhdary4915 ychauhdary4915 4096 Sep 10 04:33 .cache
drwxr-xr-x 3 ychauhdary4915 ychauhdary4915 4096 Sep 10 04:19 .config
drwxr-xr-x 3 ychauhdary4915 ychauhdary4915 4096 Sep 10 04:19 .docker
drwxr-xr-x 3 ychauhdary4915 ychauhdary4915 4096 Sep 10 04:19 .gnupg
drwxr-xr-x 5 ychauhdary4915 ychauhdary4915 4096 Sep 10 04:47 .ipython
drwxr-xr-x 2 ychauhdary4915 ychauhdary4915 4096 Sep 10 04:42 .ipyter
drwxr-xr-x 3 ychauhdary4915 ychauhdary4915 4096 Sep 10 04:33 .local
-rw-r--r-- 1 ychauhdary4915 ychauhdary4915 807 Apr 18 2019 .profile
drwxr-xr-x 2 ychauhdary4915 ychauhdary4915 4096 Sep 12 18:08 .ssh
drwxr-xr-x 3 ychauhdary4915 ychauhdary4915 4096 Sep 10 04:25 JPTR_NTBK
-rw----- 1 ychauhdary4915 ychauhdary4915 0 Sep 12 18:08 key.json

```

- We have created key.Json files found in the VM Instances home directory.

```
ssh.cloud.google.com/v2/ssh/projects/applied-abby-435016-b4/zones/us-east1-c/instances/tf2-keras-ann-vm?authuser=0&hl=en_...
ssh.cloud.google.com/v2/ssh/projects/applied-abby-435016-b4/zones/us-east1-c/instances/tf2-keras-ann-vm?authuser=0&...

SSH-in-browser UPLOAD FILE DOWNLOAD FILE
[ ] [ ] [ ]

-bash: la: command not found
(base) ychaudhary4915@tf2-keras-ann-vm:~$ ls
JPTR_NTBK key.json
(base) ychaudhary4915@tf2-keras-ann-vm:~$ ls -all
total 60
drwxr-xr-x 11 ychaudhary4915 ychaudhary4915 4096 Sep 12 18:02 .
drwxr-xr-x  6 root      root        4096 Sep 10 04:16 ..
-rw-----  1 ychaudhary4915 ychaudhary4915 490 Sep 12 18:12 .bash_history
-rw-r--r--  1 ychaudhary4915 ychaudhary4915 220 Apr 18 2019 .bash_logout
-rw-r--r--  1 ychaudhary4915 ychaudhary4915 3957 Sep 10 04:19 .bashrc
drwxr-xr-x  3 ychaudhary4915 ychaudhary4915 4096 Sep 10 04:33 .cache
drwxr-xr-x  3 ychaudhary4915 ychaudhary4915 4096 Sep 10 04:19 .config
drwxr-xr-x  2 ychaudhary4915 ychaudhary4915 4096 Sep 10 04:19 .docker
drwx-----  3 ychaudhary4915 ychaudhary4915 4096 Sep 10 04:19 .gnupg
drwxr-xr-x  5 ychaudhary4915 ychaudhary4915 4096 Sep 10 04:47 .ipython
drwxr-xr-x  2 ychaudhary4915 ychaudhary4915 4096 Sep 10 04:42 .jupyter
drwxr-xr-x  3 ychaudhary4915 ychaudhary4915 4096 Sep 10 04:33 .local
-rw-r--r--  1 ychaudhary4915 ychaudhary4915 807 Apr 18 2019 .profile
drwx-----  2 ychaudhary4915 ychaudhary4915 4096 Sep 12 18:14 .ssh
drwxr-xr-x  3 ychaudhary4915 ychaudhary4915 4096 Sep 10 04:25 JPTR_NTBK
-rw-----  1 ychaudhary4915 ychaudhary4915     0 Sep 12 18:09 key.json
(base) ychaudhary4915@tf2-keras-ann-vm:~$ 
```

- Run Command below line.

```
gcloud iam service-accounts keys create ~/key.json --iam-account service-acc-dl-tf-cnn@applied-abbe-  
435016-b4.iam.gserviceaccount.com
```

```
ssh.cloud.google.com/v2/ssh/projects/applied-abby-435016-b4/zones/us-east1-c/instances/tf2-keras-ann-vm?authuser=0&hl=en...
ssh.cloud.google.com/v2/ssh/projects/applied-abby-435016-b4/zones/us-east1-c/instances/tf2-keras-ann-vm?authuser=0...

SSH-in-browser
UPLOAD FILE DOWNLOAD FILE ⚙️ 📊 🗃️

-rw-r--r-- 1 ychaudhary4915 ychaudhary4915 807 Apr 18 2019 .profile
drwx----- 2 ychaudhary4915 ychaudhary4915 4096 Sep 12 18:17 .ssh
drwxr-xr-x 3 ychaudhary4915 ychaudhary4915 4096 Sep 10 04:25 JPTR_NTBK
-rw----- 1 ychaudhary4915 ychaudhary4915 0 Sep 12 18:17 key.json
(base) ychaudhary4915@tf2-keras-ann-vm:~$ gcloud iam service-accounts keys create ~/key.json --iam-account se
rvice-acc-dl-tf-cnn@applied-abby-435016-b4.iam.gserviceaccount.com
ERROR: (gcloud.iam.service-accounts.keys.create) PERMISSION_DENIED: Request had insufficient authentication s
copes.
- '@type': type.googleapis.com/google.rpc.ErrorInfo
  domain: googleapis.com
  metadata:
    method: google.iam.admin.v1.IAM.CreateServiceAccountKey
    service: iam.googleapis.com
  reason: ACCESS_TOKEN_SCOPE_INSUFFICIENT

If you are in a compute engine VM, it is likely that the specified scopes during VM creation are not enough t
o run this command.
See https://cloud.google.com/compute/docs/access/service-accounts#accessscopesiam for more information about a
ccess scopes.
See https://cloud.google.com/compute/docs/access/create-enable-service-accounts-for-instances#changeserviceac
countandscopes for how to update access scopes of the VM.
(base) ychaudhary4915@tf2-keras-ann-vm:~$ 
```

finally, it's done, gcp natural language api has been successfully set up.

9. PART VII: Join a Group (10 Points)

Discussion

- **Yog Chaudhary**
- **Azhan Saleem**
-

10. HOWTO Submit

The student must submit all the sections, i.e., submission requirements, in a Microsoft Word document sent to the instructor (**Thuan.Nguyen@unt.edu**) as an attachment to a UNT email.

The subject of the email must be: "**ADTA 5750: Assignment 1 – Submission.**"