Lab 1 – Running Jupyter Notebook on Google Cloud

Course: Big Data Analytics
Instructor: David Johnson
Student: Ruth Ihunanya Chimezuru Obere
Date: 10 November 2025

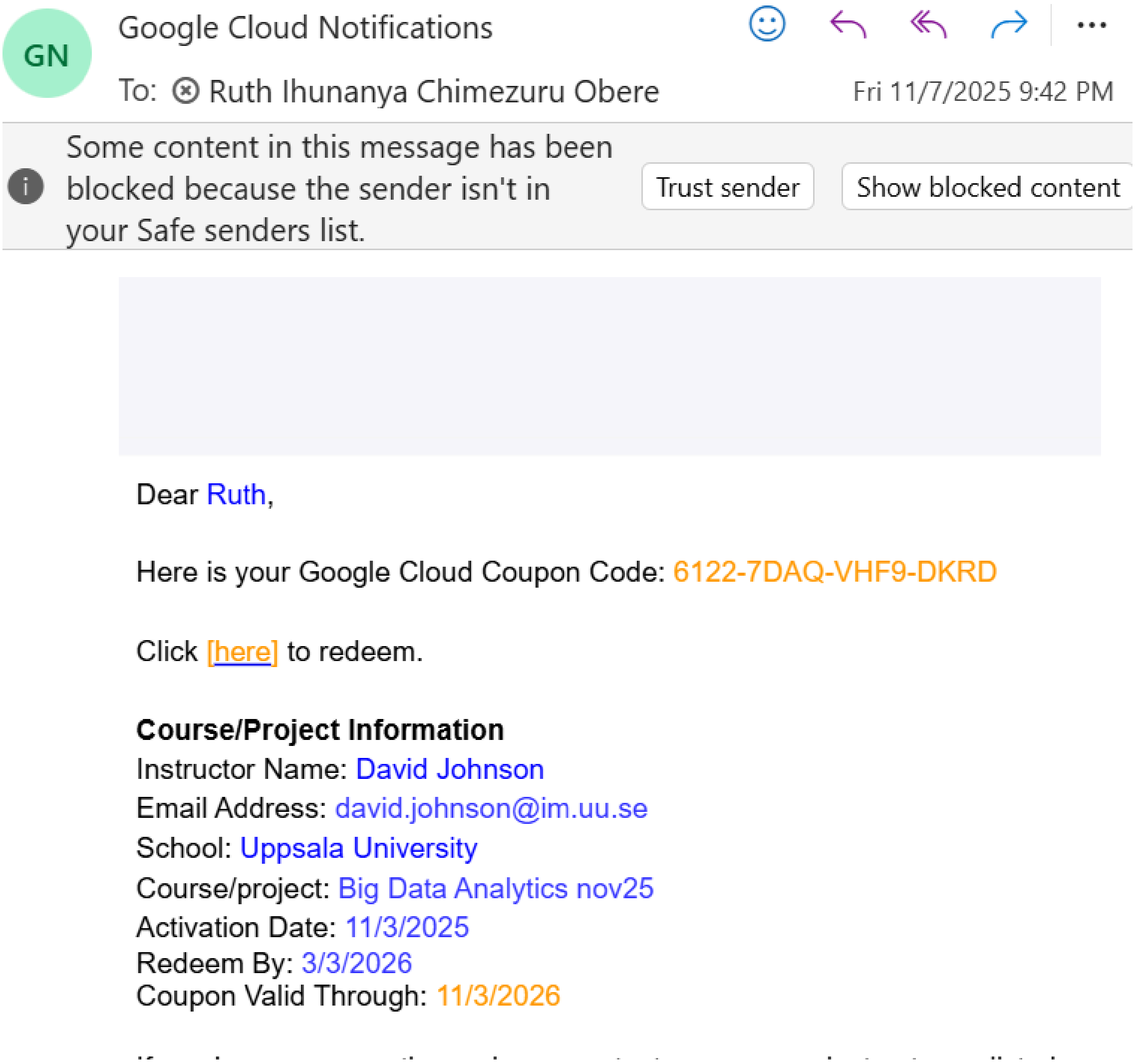
Objective

To set up a Google Cloud environment, create a virtual machine (VM), install Jupyter Notebook on it, and connect to the notebook server from my computer.

Step 1: Access Google Cloud Console and Redeem Coupon

After following the instructions given by the lecturer, I opened the **Google Cloud Console**.
A verification email appeared, and after completing the process, a **coupon code** was sent to my school email.

I redeemed the coupon successfully, which gave me free Google Cloud credits to use for this course.
Once access was confirmed, I continued to create a server instance where Jupyter Notebook would run.



Google Cloud Notifications

To: Ruth Ihunanya Chimezuru Obere

Fri 11/7/2025 9:42 PM

Some content in this message has been blocked because the sender isn't in your Safe senders list.

Dear Ruth,

Here is your Google Cloud Coupon Code: 6122-7DAQ-VHF9-DKRD

Click [here] to redeem.

Course/Project Information
Instructor Name: David Johnson
Email Address: david.johnson@im.uu.se
School: Uppsala University
Course/project: Big Data Analytics nov25
Activation Date: 11/3/2025
Redeem By: 3/3/2026
Coupon Valid Through: 11/3/2026

Step 2: Create a Virtual Machine (VM) Instance

1. Logged into **Google Cloud Console** using my **school email** and the **coupon code** received via email, which provided free credits for this course.

2. Went to **Compute Engine** → **VM Instances** → **Create Instance**.

3. Since this was my first time using Compute Engine, a page appeared about the **Compute Engine API**.

   - Clicked **Enable** to activate the API.

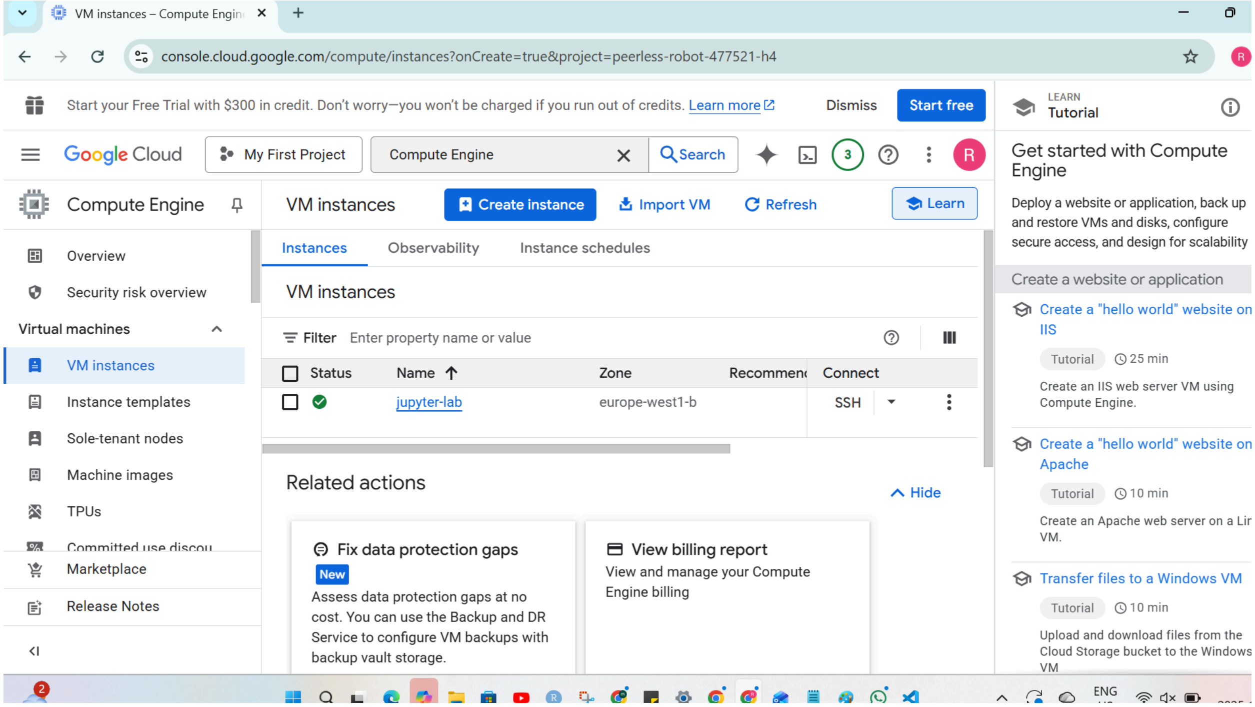
   - After a short wait, I was redirected to the **Create Instance** page.

4. Configured the VM as follows:

   - **Name:** `jupyter-lab`
   - **Region/Zone:** nearest to my location (e.g., `europe-west1-b`)
   - **Machine Type:** `e2-medium (2 vCPUs, 4 GB RAM)`
   - **CPU Architecture:** `AMD64 (x86_64)`
   - **Operating System:** `Ubuntu 22.04 LTS (Linux)`
   - **Boot Disk Size:** 10 GB balanced persistent disk (default)
   - **Firewall:** Checked both **Allow HTTP traffic** and **Allow HTTPS traffic**
   - **Load Balancer Health Checks:** left unchecked

5. Clicked **Create** to launch the instance.

6. Once created, the instance appeared in the **VM list** on my dashboard



Start your Free Trial with $300 in credit. Don't worry—you won't be charged if you run out of credits. Learn more

Status	Name	Zone	Recommend	Connect
✓	jupyter-lab	europe-west1-b		SSH

Fix data protection gaps
New
Assess data protection gaps at no cost. You can use the Backup and DR Service to configure VM backups with backup vault storage.

View billing report
View and manage your Compute Engine billing

Deploy a website or application, back up and restore VMs and disks, configure secure access, and design for scalability

Create an IIS web server VM using Compute Engine.

Create an Apache web server on a Lir VM.

Upload and download files from the Cloud Storage bucket to the Windows VM

Step 3: Connect to the Virtual Machine via SSH

After creating the VM instance, I connected to it using the **SSH option** in the Google Cloud Console.

1. From the **VM Instances** dashboard, I located my instance named **jupyter-lab**.

2. Under the **Connect** column, I clicked the dropdown arrow beside **SSH** and selected **"Open in browser window."**

   At first, the connection **failed**, and I got the following error message:

   "Connection via Cloud Identity-Aware Proxy Failed – Code: 4003, Reason: failed to connect to backend."

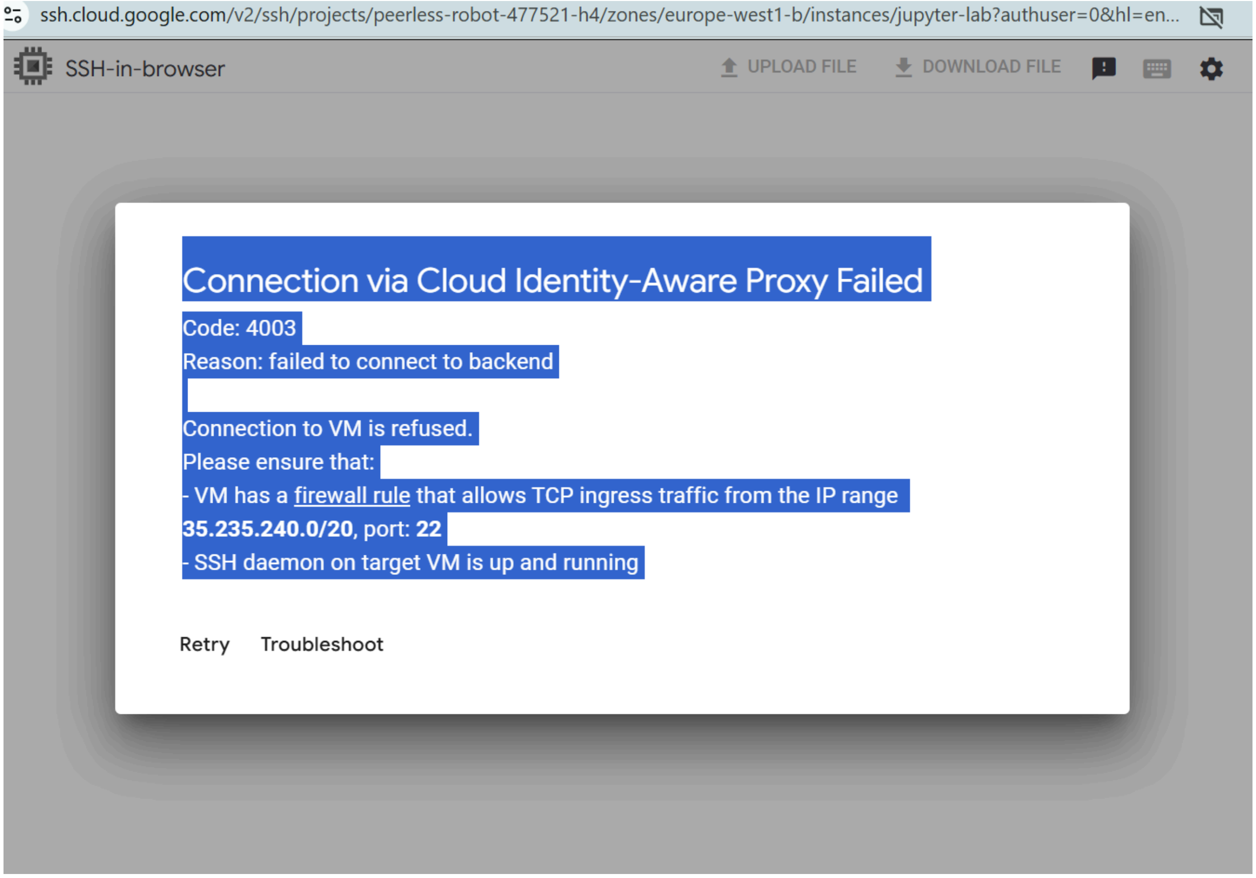
3. The error message also said that the connection to the VM was being refused and suggested checking the **firewall rules** to allow TCP ingress traffic from IP range `35.235.240.0/20` on **port 22**.

4. To fix this, I went to **VPC Network** → **Firewall**, created a new firewall rule, and allowed **TCP traffic on port 22**.

5. After saving the rule, I went back to **VM Instances** and tried connecting again. This time, the SSH connection was successful, and a terminal window opened in the browser showing the Ubuntu prompt:

   ruth_ihunanya_chimezuru_obere_59@jupyter-lab:~$

6. This confirmed that I had successfully connected to my virtual machine through a secure SSH session and could proceed with installing Jupyter Notebook.


ssh.cloud.google.com/v2/ssh/projects/peerless-robot-477521-h4/zones/europe-west1-b/instances/jupyter-lab?authuser=0&hl=en...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

Connection via Cloud Identity-Aware Proxy Failed

Code: 4003

Reason: failed to connect to backend

Connection to VM is refused.

Please ensure that:

- VM has a firewall rule that allows TCP ingress traffic from the IP range **35.235.240.0/20**, port: **22**

- SSH daemon on target VM is up and running

Step 4: Install Jupyter Notebook on the VM

Once connected to the VM via SSH, I installed Jupyter Notebook to run Python notebooks on the cloud.

1. **Update the system packages**
   I Ran the following commands:

   ```
   sudo apt update

   sudo apt upgrade -y
   ```

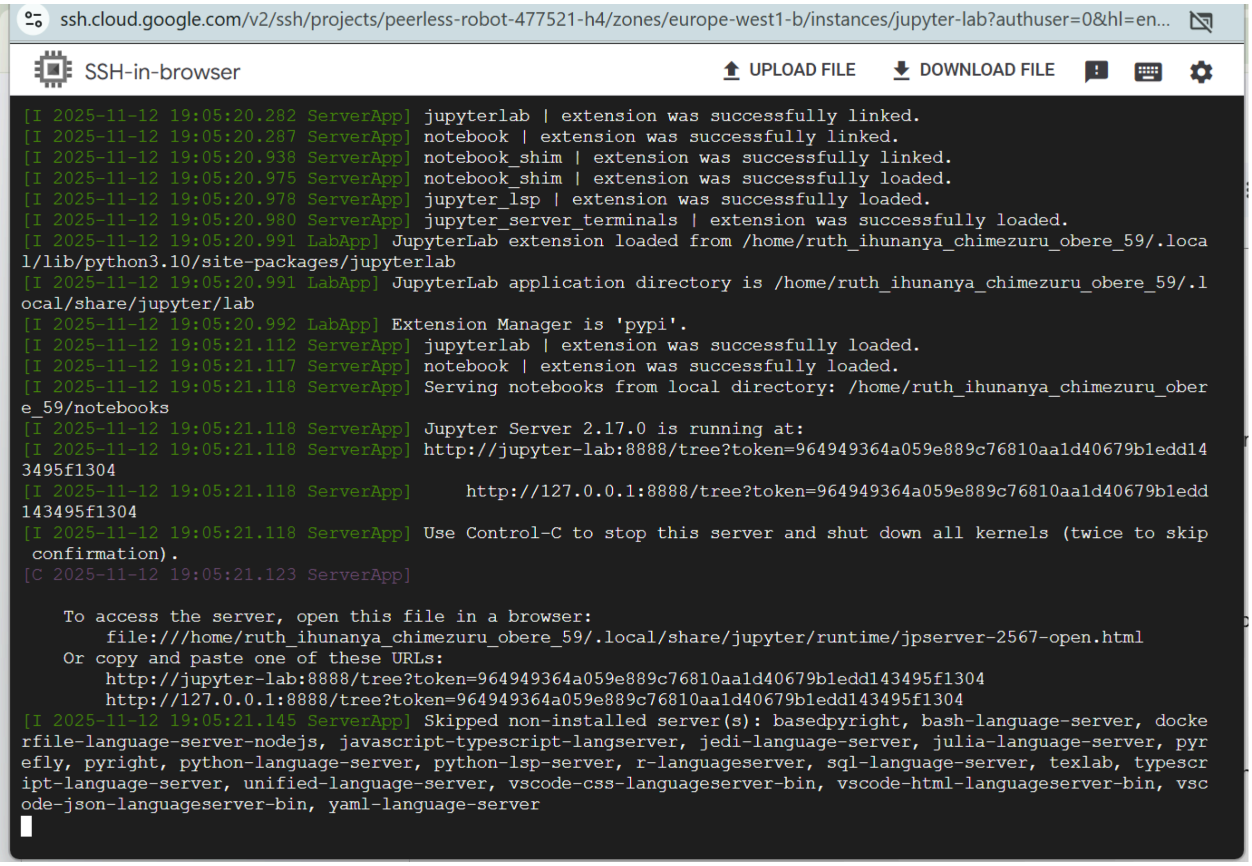
2. This updated all packages to the latest version. The upgrade took a few minutes.
   Install Python and pip
   Ensured Python and pip were installed:

   ```
   sudo apt install python3 python3-pip -y
   ```

3. **Install Jupyter Notebook**
   Installed Jupyter using pip:

   ```
   pip install notebook
   ```

4. **Start Jupyter Notebook server**
   I ran: `jupyter notebook --ip=0.0.0.0 --port=8888 --no-browser` This started the server and displayed a URL containing a **token**, which is needed to access Jupyter in the browser.



ssh.cloud.google.com/v2/ssh/projects/peerless-robot-477521-h4/zones/europe-west1-b/instances/jupyter-lab?authuser=0&hl=en...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

```
[I 2025-11-12 19:05:20.282 ServerApp] jupyterlab | extension was successfully linked.
[I 2025-11-12 19:05:20.287 ServerApp] notebook | extension was successfully linked.
[I 2025-11-12 19:05:20.938 ServerApp] notebook_shim | extension was successfully linked.
[I 2025-11-12 19:05:20.975 ServerApp] notebook_shim | extension was successfully loaded.
[I 2025-11-12 19:05:20.978 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2025-11-12 19:05:20.980 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2025-11-12 19:05:20.991 LabApp] JupyterLab extension loaded from /home/ruth_ihunanya_chimezuru_obere_59/.loca
l/lib/python3.10/site-packages/jupyterlab
[I 2025-11-12 19:05:20.991 LabApp] JupyterLab application directory is /home/ruth_ihunanya_chimezuru_obere_59/.l
ocal/share/jupyter/lab
[I 2025-11-12 19:05:20.992 LabApp] Extension Manager is 'pypi'.
[I 2025-11-12 19:05:21.112 ServerApp] jupyterlab | extension was successfully loaded.
[I 2025-11-12 19:05:21.117 ServerApp] notebook | extension was successfully loaded.
[I 2025-11-12 19:05:21.118 ServerApp] Serving notebooks from local directory: /home/ruth_ihunanya_chimezuru_ober
e_59/notebooks
[I 2025-11-12 19:05:21.118 ServerApp] Jupyter Server 2.17.0 is running at:
[I 2025-11-12 19:05:21.118 ServerApp] http://jupyter-lab:8888/tree?token=964949364a059e889c76810aa1d40679b1edd14
3495f1304
[I 2025-11-12 19:05:21.118 ServerApp]     http://127.0.0.1:8888/tree?token=964949364a059e889c76810aa1d40679b1edd
143495f1304
[I 2025-11-12 19:05:21.118 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip
 confirmation).
[C 2025-11-12 19:05:21.123 ServerApp]

    To access the server, open this file in a browser:
        file:///home/ruth_ihunanya_chimezuru_obere_59/.local/share/jupyter/runtime/jpserver-2567-open.html
    Or copy and paste one of these URLs:
        http://jupyter-lab:8888/tree?token=964949364a059e889c76810aa1d40679b1edd143495f1304
        http://127.0.0.1:8888/tree?token=964949364a059e889c76810aa1d40679b1edd143495f1304
[I 2025-11-12 19:05:21.145 ServerApp] Skipped non-installed server(s): basedpyright, bash-language-server, docke
rfile-language-server-nodejs, javascript-typescript-langserver, jedi-language-server, julia-language-server, pyr
efly, pyright, python-language-server, python-lsp-server, r-languageserver, sql-language-server, texlab, typescr
ipt-language-server, unified-language-server, vscode-css-languageserver-bin, vscode-html-languageserver-bin, vsc
ode-json-languageserver-bin, yaml-language-server
```

Step 5: Start Jupyter Notebook on the VM

After creating the VM, I connected to it using SSH through the Google Cloud Console.

1. **Start the Jupyter Notebook server**
   I ran the command:
   jupyter notebook --ip=0.0.0.0 --port=8888 --no-browser

2. **Initial errors**
   At first, I received an error saying:
   jupyter: command not found

This was because the Jupyter Notebook had not been installed yet. After installing Jupyter (see Step 4), I retried the command successfully.

3. **Accessing Jupyter Notebook**
   The terminal displayed the Jupyter Notebook server running, along with a token URL, for example:

   http://127.0.0.1:8888/?token=<TOKEN>

4. **Connection issue in browser**
   When I tried to open the notebook in my browser using the VM's external IP (104.155.20.179), I got:
   ERR_CONNECTION_TIMED_OUT

This happened because the firewall rules did not allow incoming traffic on port 8888.

5. **Fixing the firewall**
   To resolve this, I created a firewall rule in Google Cloud with the following settings:

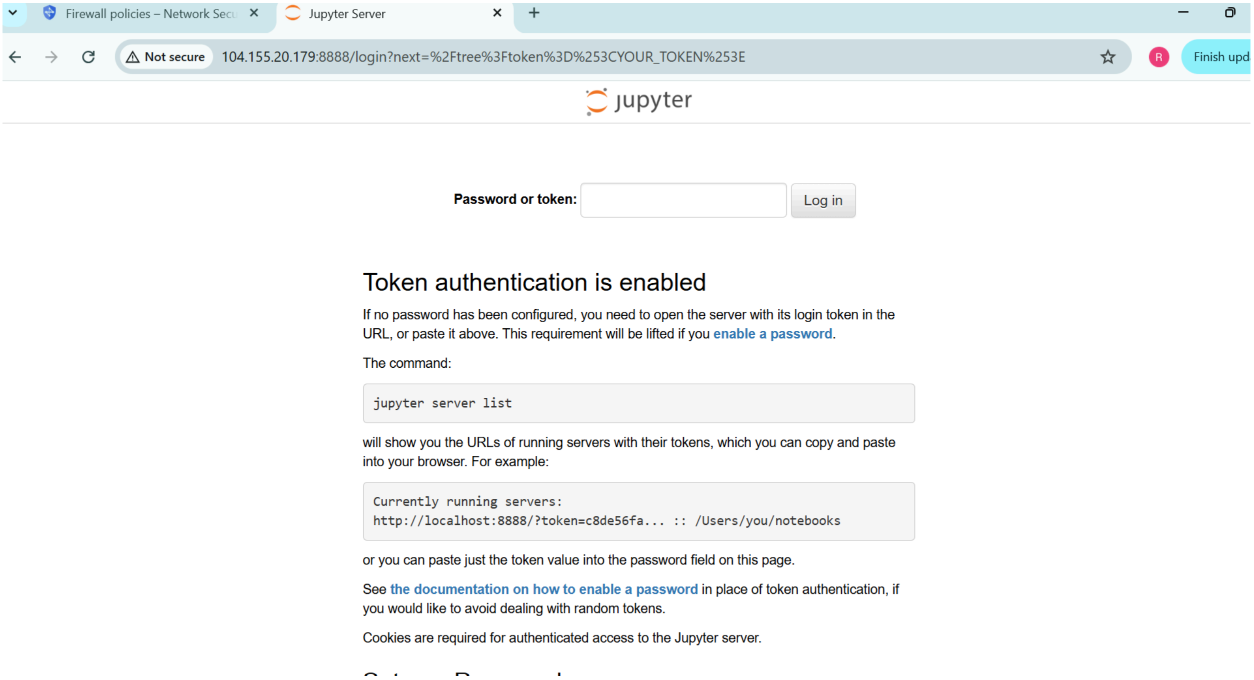
- **Name:** allow-jupyter
- **Direction:** Ingress (incoming traffic)
- **Action:** Allow
- **Targets:** All instances in the network
- **Source IP ranges:** 0.0.0.0/0
- **Protocols and ports:** tcp:8888

After creating the firewall rule, I refreshed the browser and successfully opened Jupyter Notebook at:

http://104.155.20.179:8888/?token=<TOKEN>

Challenges faced:

- SSH authentication delays initially.
- Jupyter command not found until installed.
- Browser could not connect until the firewall rule was set up.


Firewall policies – Network Secu
Jupyter Server
Not secure
104.155.20.179:8888/login?next=%2Ftree%3Ftoken%3D%253CYOUR_TOKEN%253E
Finish upd
jupyter
Password or token:
Log in

Token authentication is enabled

If no password has been configured, you need to open the server with its login token in the URL, or paste it above. This requirement will be lifted if you **enable a password**.

The command:

```
jupyter server list
```

will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:

```
Currently running servers:
http://localhost:8888/?token=c8de56fa... :: /Users/you/notebooks
```

or you can paste just the token value into the password field on this page.

See **the documentation on how to enable a password** in place of token authentication, if you would like to avoid dealing with random tokens.

Cookies are required for authenticated access to the Jupyter server.

Step 6: Access Jupyter Notebook on the VM

After starting the Jupyter Notebook server on the VM, the terminal displayed a URL containing a **token**, like:

http://127.0.0.1:8888/?token=<LONG_STRING>

1. I copied the **token** part from the URL (everything after `token=`).
2. I opened a browser on my computer.
3. I typed my VM's external IP and port in the browser, like this:

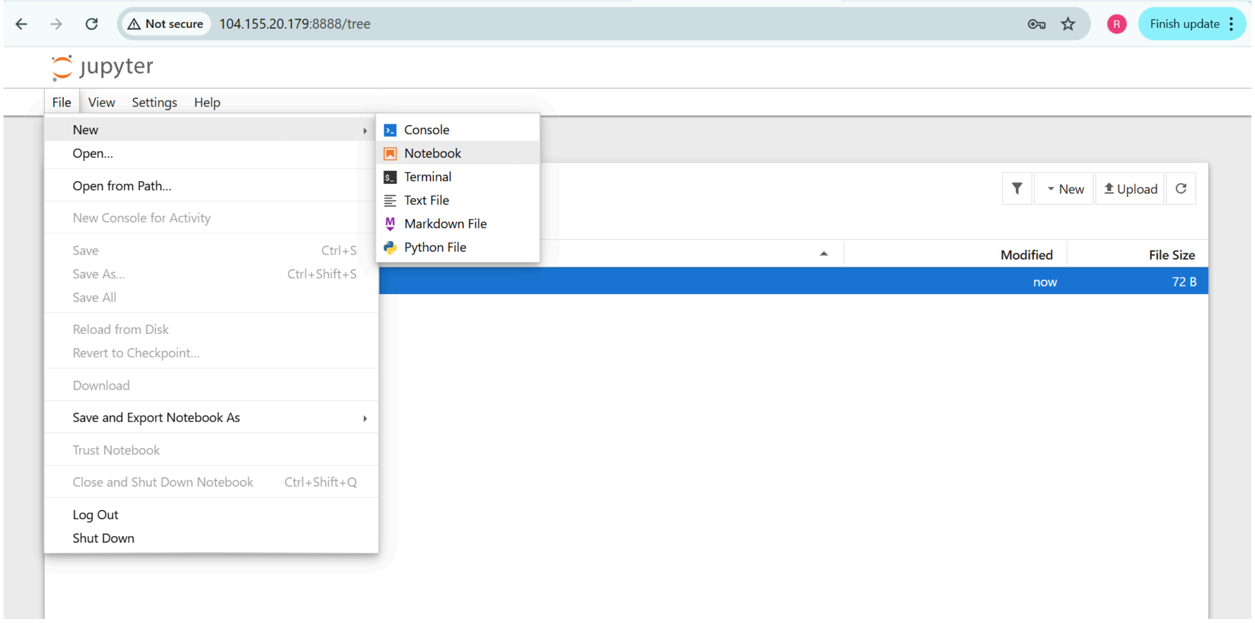
   http://104.155.20.179:8888
4. When prompted, I **pasted the token** into the login field.
5. This successfully opened the Jupyter Notebook in my browser, showing the interface where I can create new notebooks and write Python code.

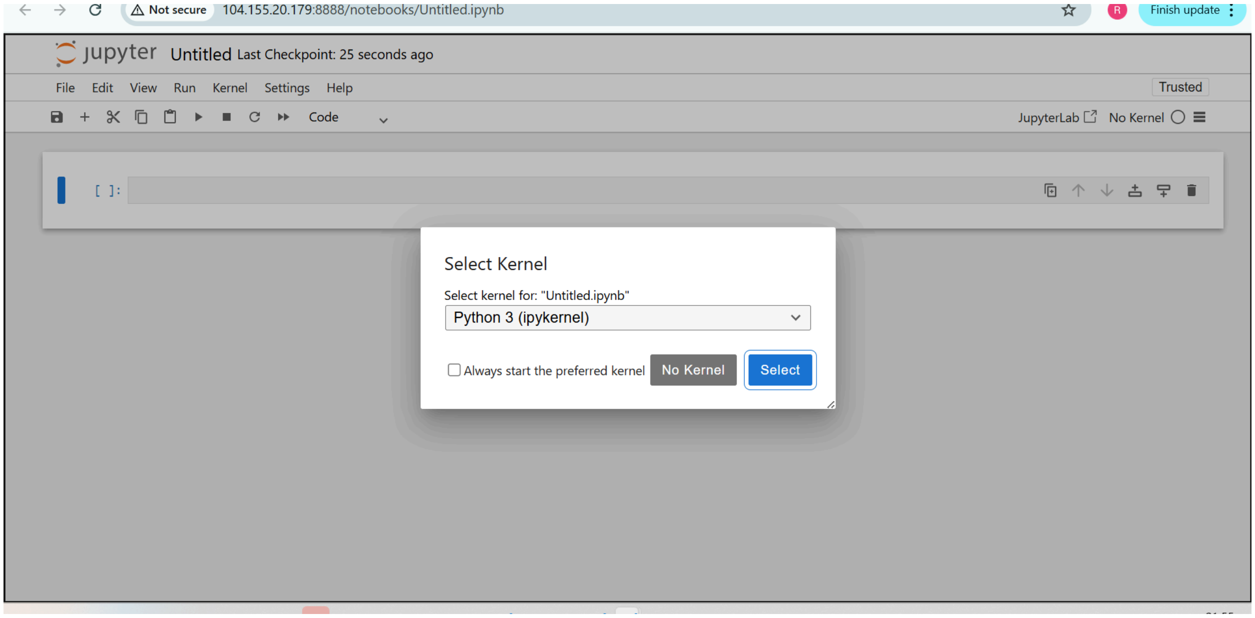
Challenges faced:

- Initially, the browser could not connect until the firewall rule allowing traffic on port 8888 was created.
- Copying the token correctly was important to log in.

Outcome:

I can now fully access Jupyter Notebook on the cloud from my local browser and start running Python code.





Select Kernel

Select kernel for: "Untitled.ipynb"

Python 3 (ipykernel)

□ Always start the preferred kernel

Conclusion

In this lab, I successfully set up a virtual machine (VM) on Google Cloud Platform and installed a Jupyter Notebook to run Python code in the cloud. I encountered a few challenges along the way:

- ➔ Initially, SSH authentication took some time to connect.
- ➔ The `jupyter` command was not found until I installed the Jupyter Notebook on the VM.
- ➔ My browser could not connect to the Jupyter Notebook server until I created a firewall rule to allow incoming traffic on port 8888.

After resolving these issues:

- ➔ I was able to access the Jupyter Notebook using the token provided in the terminal.
- ➔ I set up a password for easier access in future sessions.
- ➔ Ubuntu proved easier for installations compared to Windows.

- ➔ Allowing HTTP/HTTPS traffic was necessary to open Jupyter in the browser.

Overall, the setup was successful, and the notebook environment is ready for running Python code and performing analytics on large datasets.