5. Riva Server

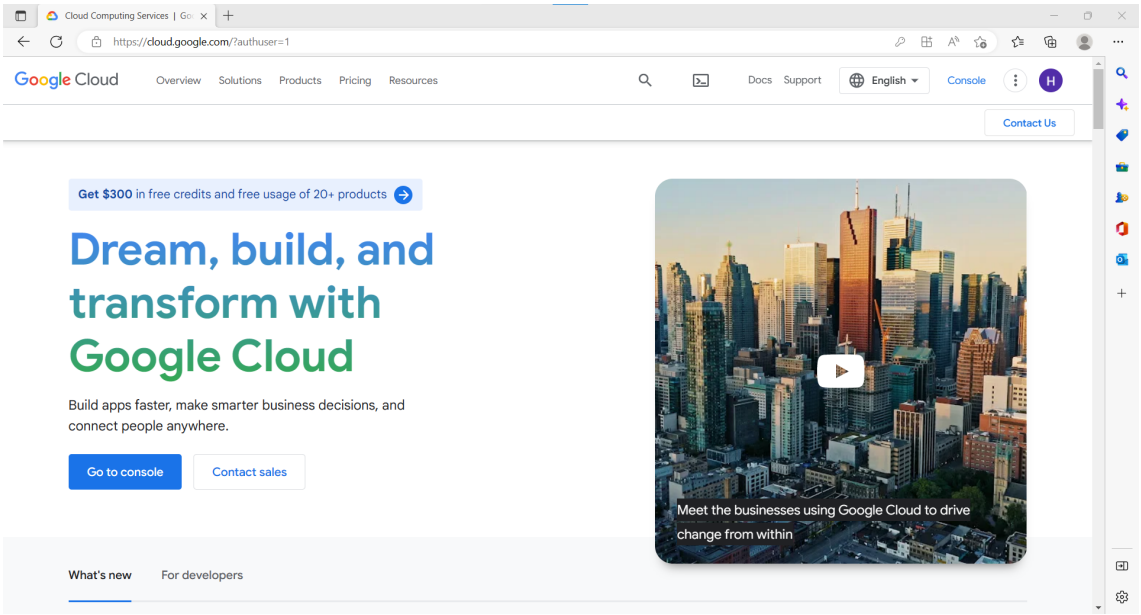
5.1 Background

There are many choices of Riva Server Deployment, some examples are AWS and Google Cloud Service. In this tutorial, we will be focusing on deploying Riva Server using the Compute Engine on Google Cloud Service.

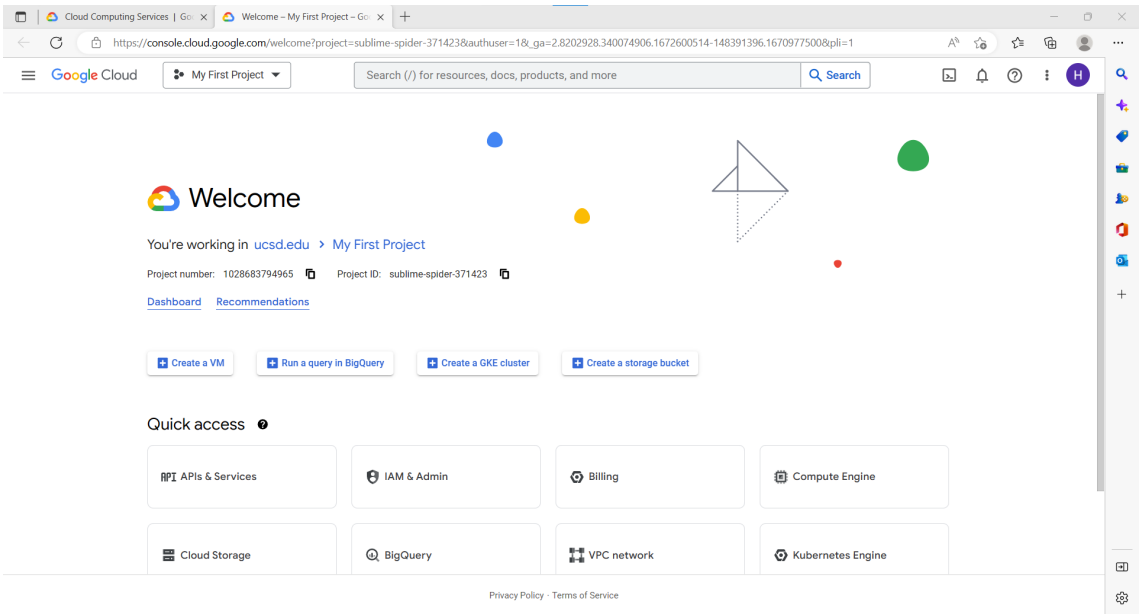
5.2 Server Setup

5.2.1 Google Cloud Server

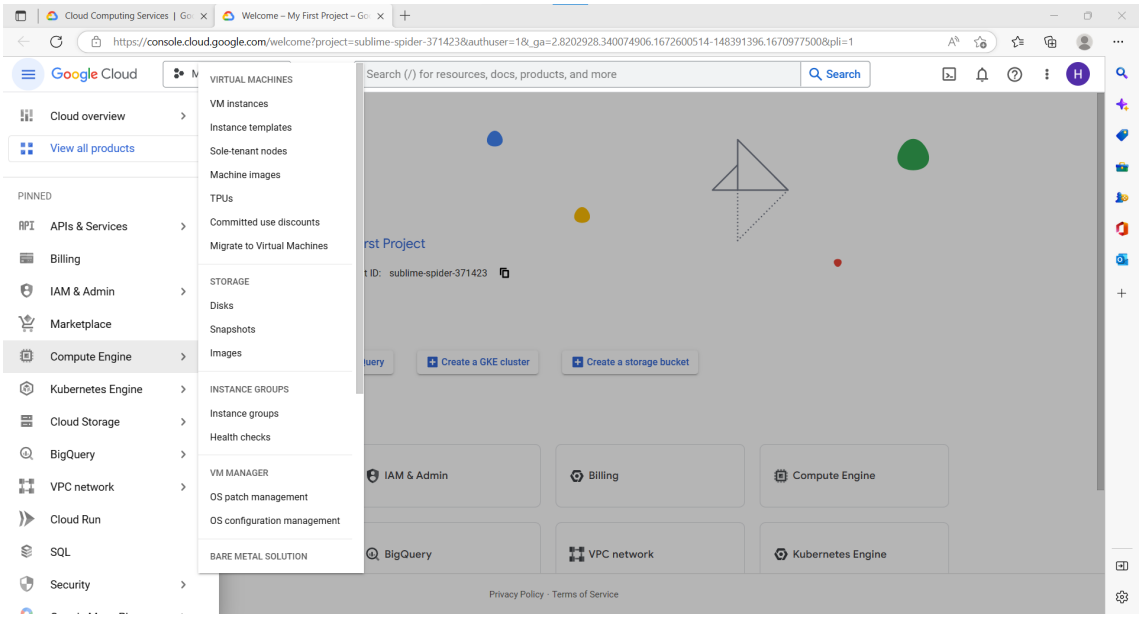
1. Go to `https://console.cloud.google.com/`.
2. Click on the **Go to console** symbol (if applicable).



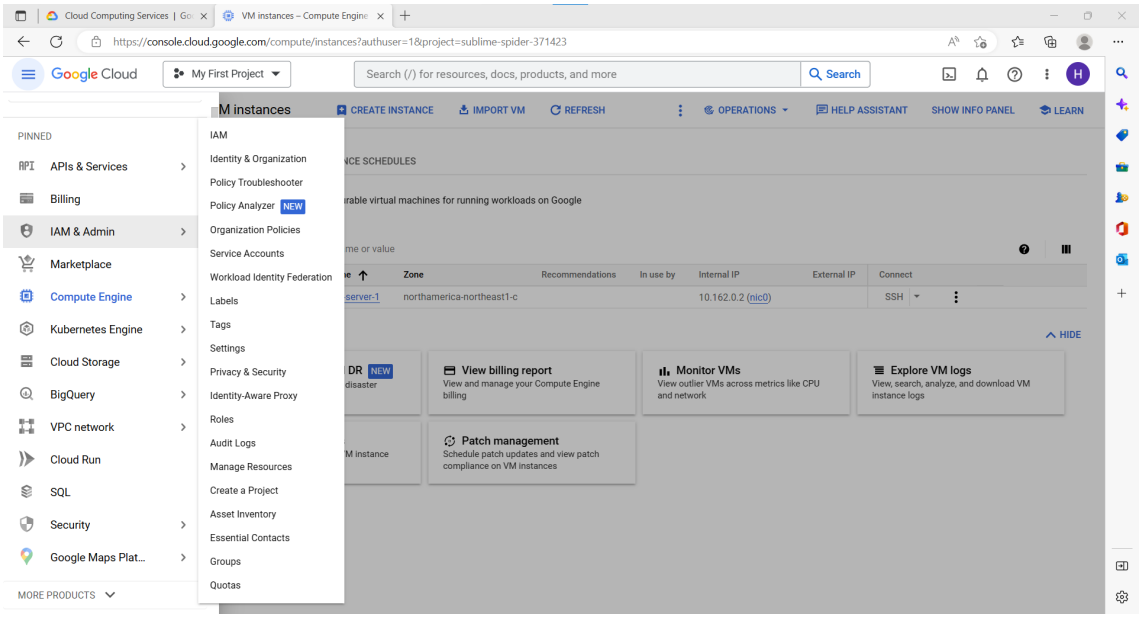
3. Click on the *3-hyphen symbol*.



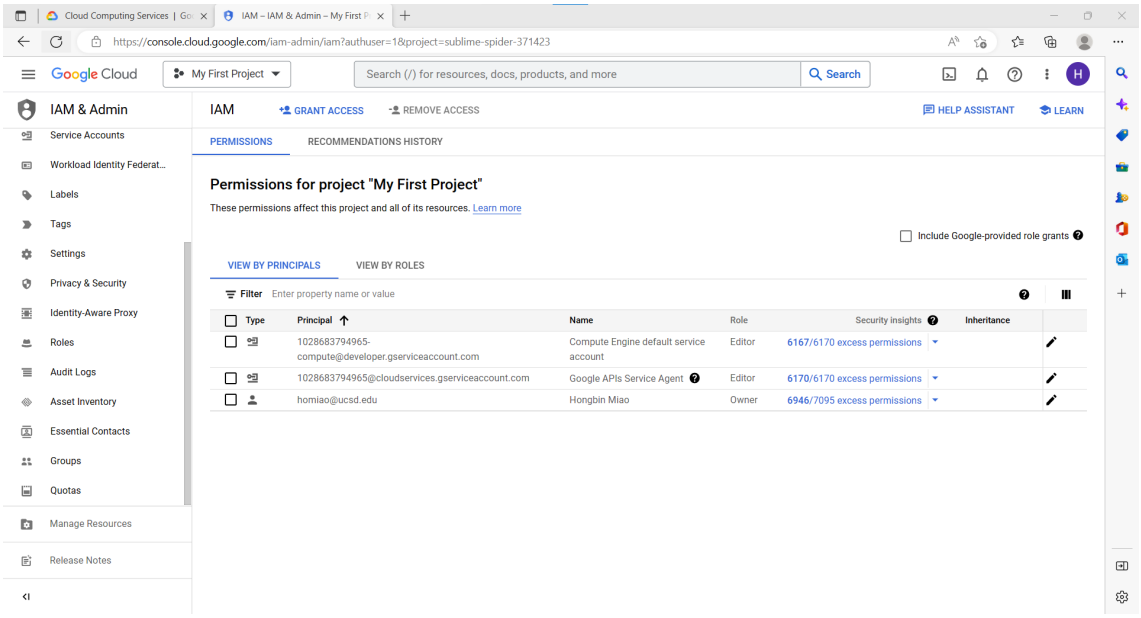
4. Go to the **Compute Engine** section.



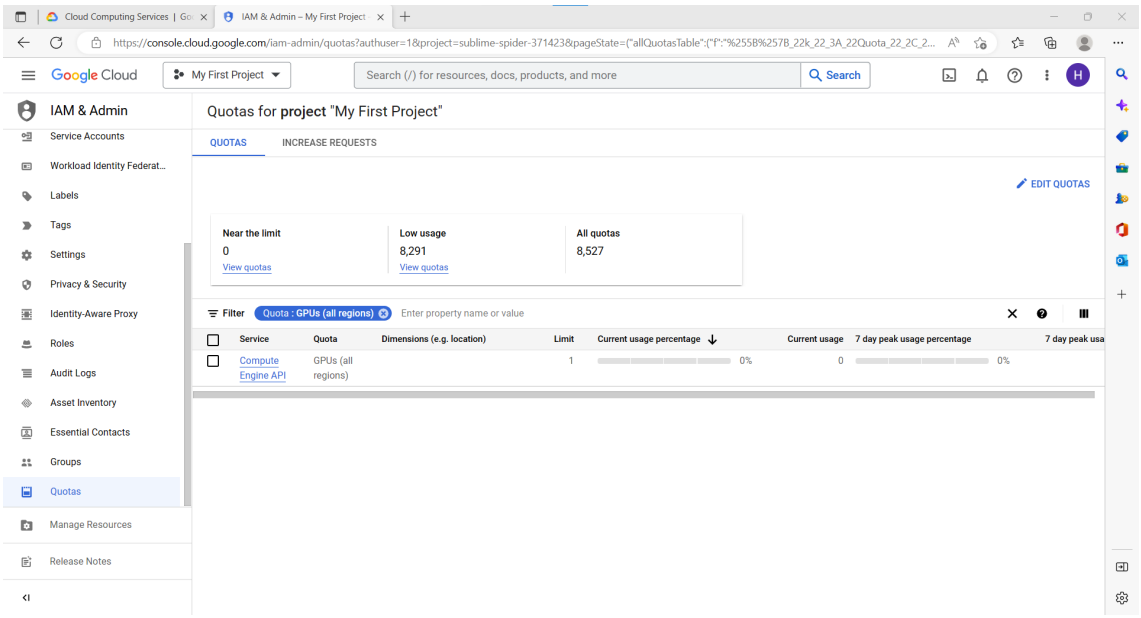
5. Activate the **Compute Engine**.
6. Click on the *3-hyphen symbol* and then click on **IAM & Admin**.



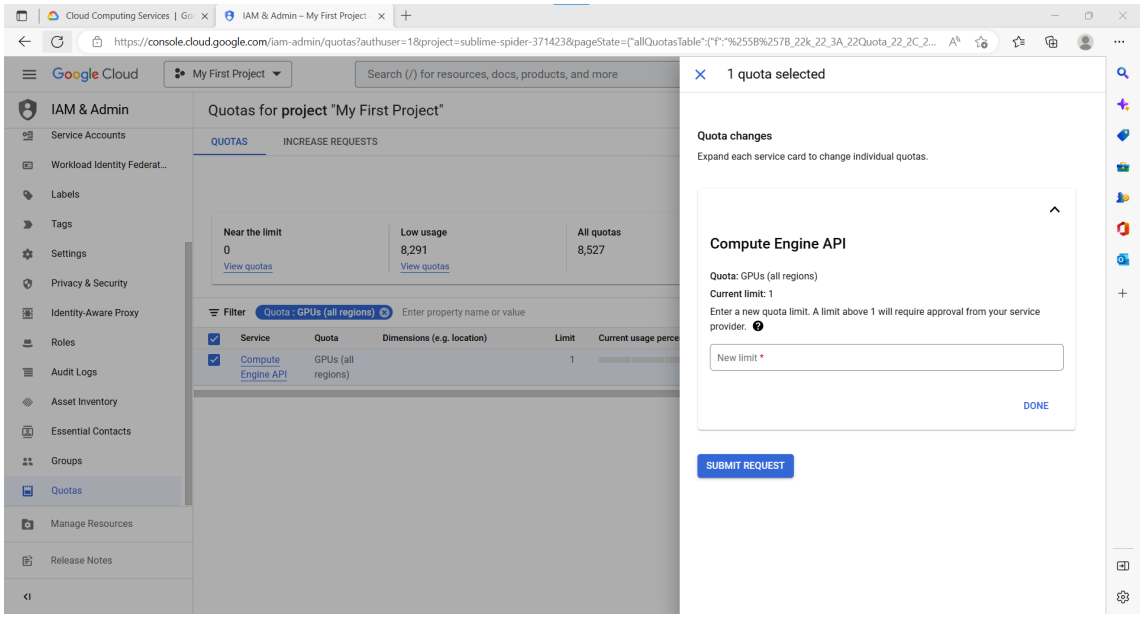
7. Within **IAM & Admin**, click on **Quotas** (since we want to have GPU access).



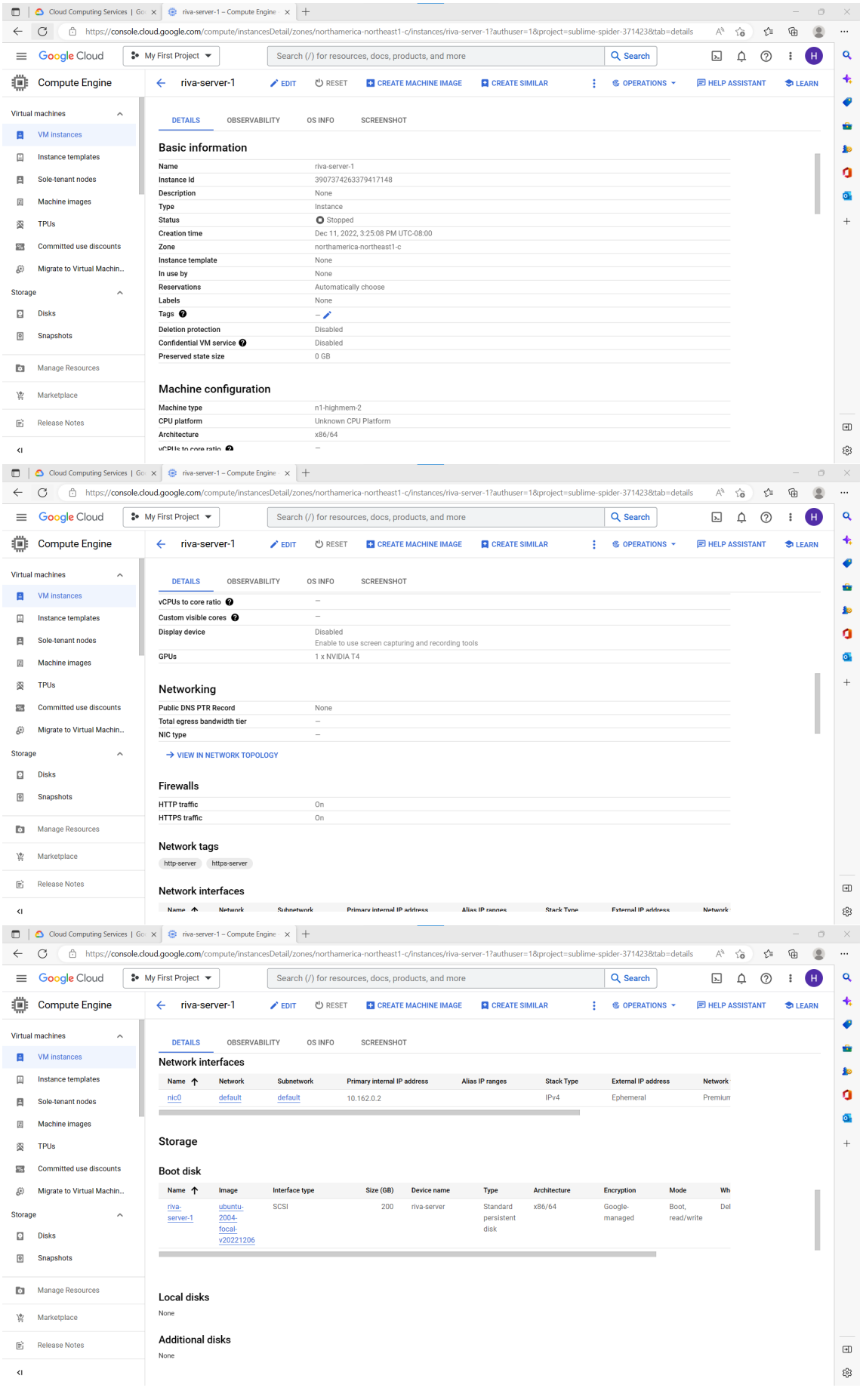
8. Inside of **Quotas**, filter the result using **GPUs (all regions)**.
9. You will then get 1 result as **Compute Engine API**; click on it and clock on the **EDIT QUOTAS** on the top right side.



10. Inside of the **EDIT QUOTAS** tab, set a new limit for your GPU quota (for my case: 1).



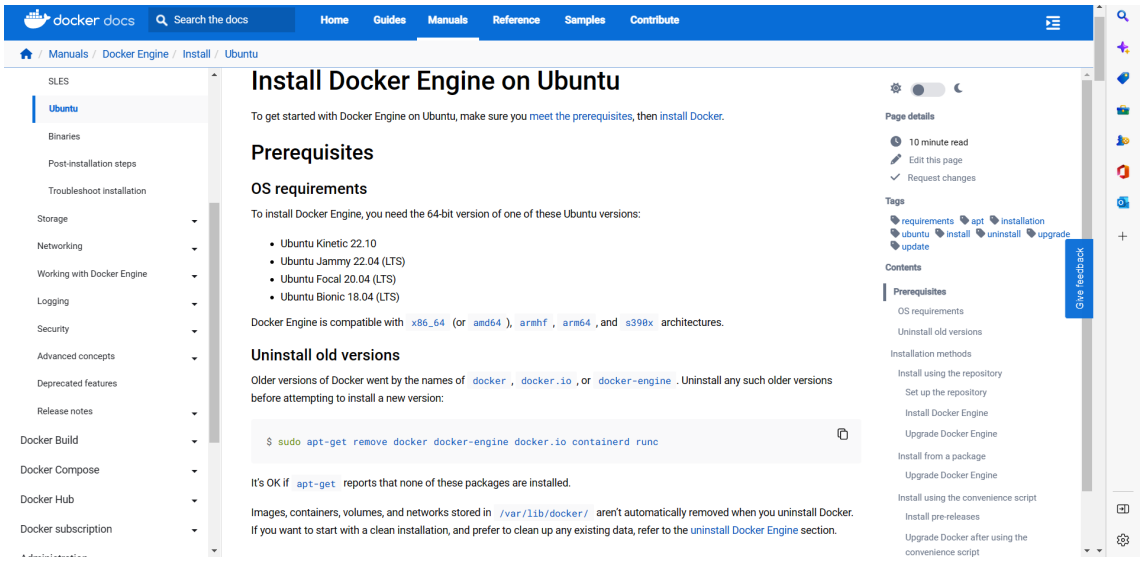
11. After you **received email about GPU quota increase**, you can proceed with the steps below.
12. Go back to **Compute Engine** section and select on **CREATE INSTANCE** (the blue tab).
13. It is time to configured your VM instance, below will be my sample configuration.


Cloud Computing Services | Go
riva-server-1 – Compute Engine
https://console.cloud.google.com/compute/instancesDetail/zones/northamerica-northeast1-c/instances/riva-server-1?authuser=1&project=sublime-spider-371423&tab=details
Google Cloud
My First Project
Search (/) for resources, docs, products, and more
Search
Compute Engine
riva-server-1
EDIT
RESET
CREATE MACHINE IMAGE
CREATE SIMILAR
OPERATIONS
HELP ASSISTANT
LEARN
Virtual machines
VM instances
Instance templates
Sole-tenant nodes
Machine images
TPUs
Committed use discounts
Migrate to Virtual Machin...
Storage
Disks
Snapshots
Manage Resources
Marketplace
Release Notes
DETAILS
OBSERVABILITY
OS INFO
SCREENSHOT
Basic information
Name riva-server-1
Instance Id 3907374263379417148
Description None
Type Instance
Status Stopped
Creation time Dec 11, 2022, 3:25:08 PM UTC-08:00
Zone northamerica-northeast1-c
Instance template None
In use by None
Reservations Automatically choose
Labels None
Tags –
Deletion protection Disabled
Confidential VM service Disabled
Preserved state size 0 GB
Machine configuration
Machine type n1-highmem-2
CPU platform Unknown CPU Platform
Architecture x86/64
vCPUs to core ratio –
Custom visible cores –
Display device Disabled
Enable to use screen capturing and recording tools
GPUs 1 x NVIDIA T4
Networking
Public DNS PTR Record None
Total egress bandwidth tier –
NIC type –
VIEW IN NETWORK TOPOLOGY
Firewalls
HTTP traffic On
HTTPS traffic On
Network tags
http-server
https-server
Network interfaces
Name Network Subnetwork Primary internal IP address Alias IP ranges Stack Type External IP address Network
nic0 default default 10.162.0.2 IPv4 Ephemeral Premium
Storage
Boot disk
Name Image Interface type Size (GB) Device name Type Architecture Encryption Mode
riva-server-1 ubuntu-2004-focal-v20221206 SCSI 200 riva-server Standard persistent disk x86/64 Google-managed Boot, read/write
Local disks
None
Additional disks
None

14. Configure your VM with similar setup, notice that you will need at least 1 GPU with the equal or greater performance to a Nvidia T4 to meet the configuration requirement of the Riva Server.
15. After the VM has been configured and started, install the Nvidia CUDA and related drivers by following the instruction on this link: https://cloud.google.com/compute/docs/gpus/install-drivers-gpu

5.2.2 Docker

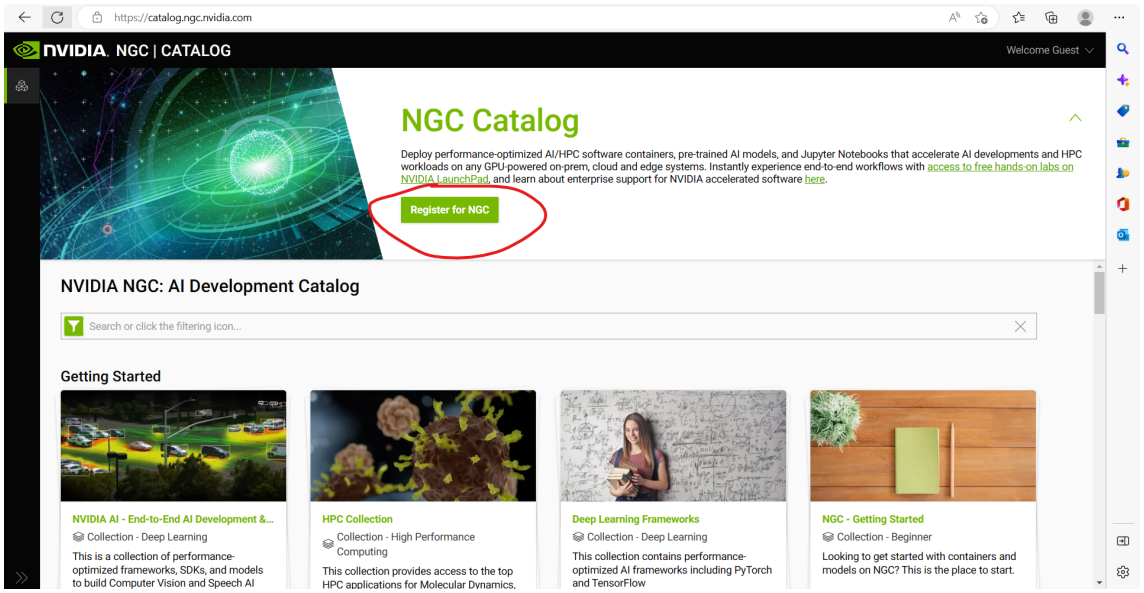
1. Install docker by following the official tutorial: `https://docs.docker.com/engine/install/ubuntu/`.



2. Install the Nvidia Docker Container by following the instruction in this link for Ubuntu 20.04: `https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/install-guide.html#docker`.

5.2.3 Nvidia NGC CLI

1. Register an account in `https://catalog.ngc.nvidia.com/`



2. SSH into the Google Cloud Server with root user access (type sudo -i in the SSH terminal), then type the command by the picture below:

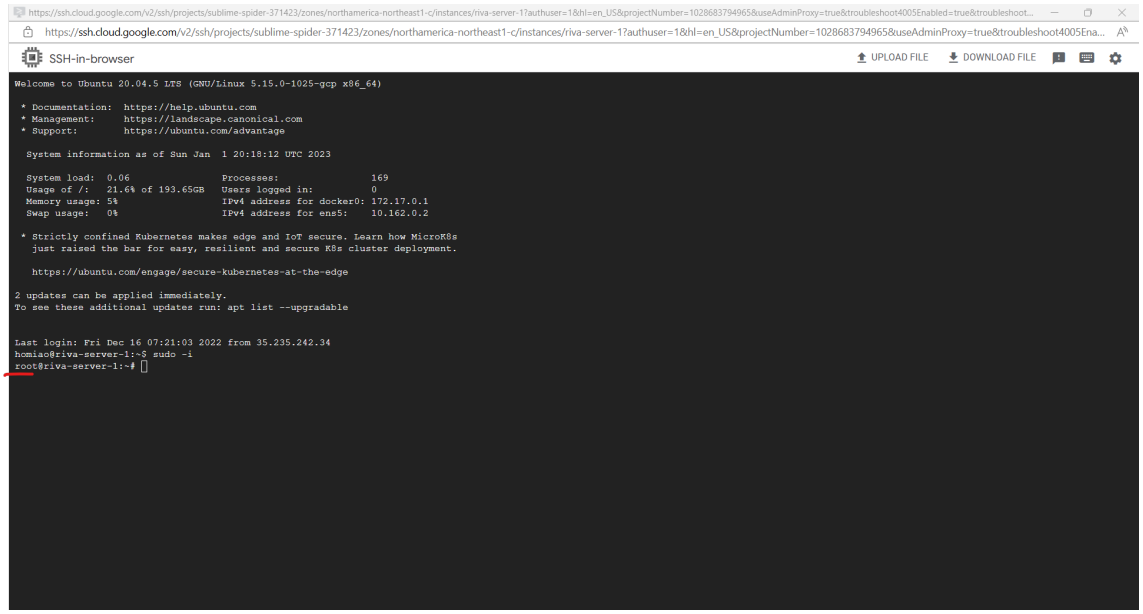
```
wget -o ngccli_cat_linux.zip https://ngc.nvidia.com/downloads/ngccli_cat_linux.zip

unzip -o ngccli_cat_linux.zip

find ngc-cli/ -type f -exec md5sum {} + | LC_ALL=C sort | md5sum -c ngc-cli.md5

echo "export PATH=\"\$PATH:$(pwd)/ngc-cli\"" >> ~/.bash_profile && source ~/.bash_profile

ngc config set
```

5.2.4 Riva SDK

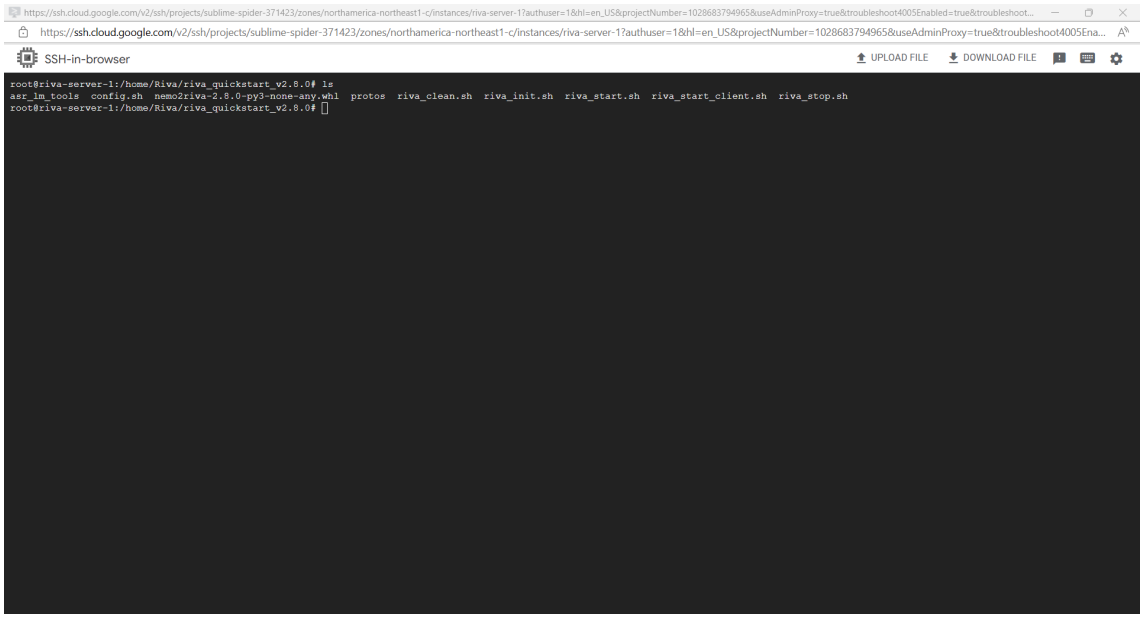
Prerequisites

- Set up Google Cloud Server
- Set up Docker
- Set up Nvidia NGC CLI

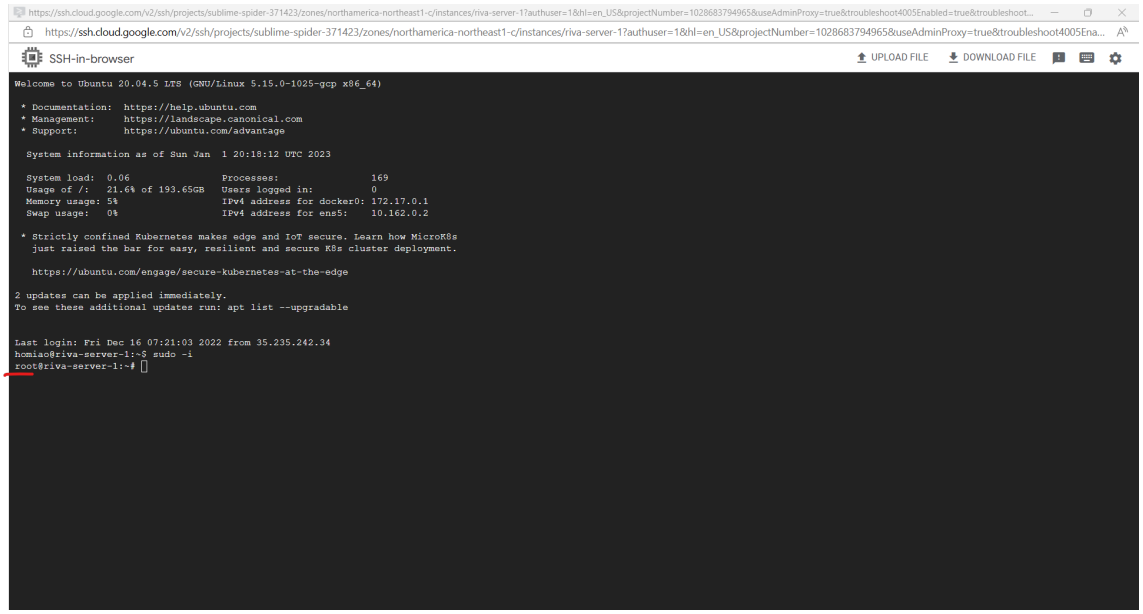
1. SSH into your Riva Server, and make sure you log in the terminal as root user by typing **sudo -i**.



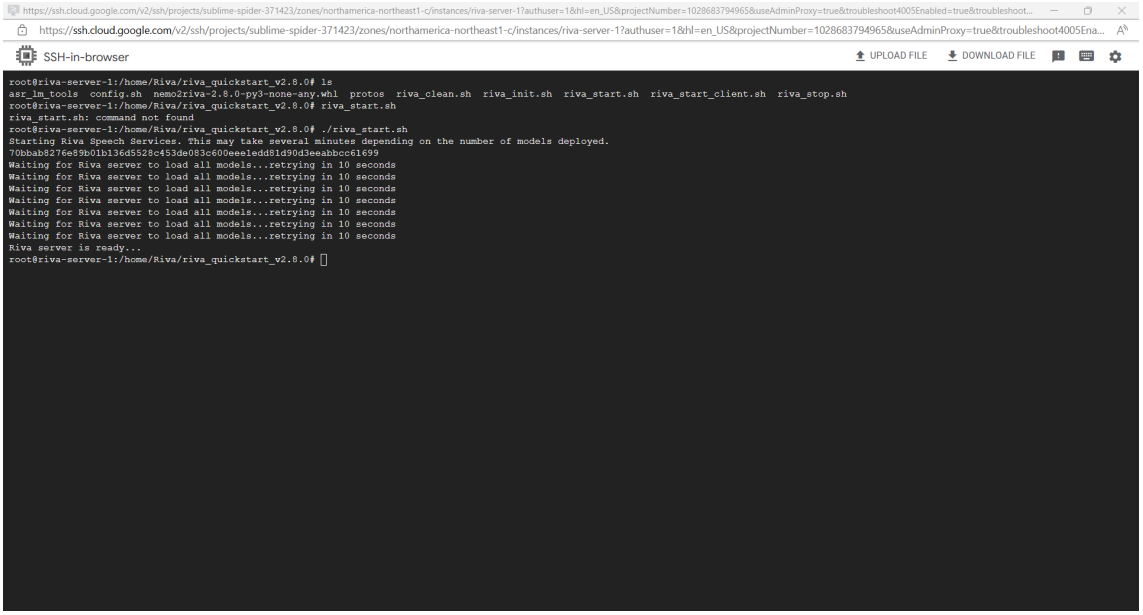
2. With the NGC CLI already set up, download the Riva Skills Quick Start through this link: `https://catalog.ngc.nvidia.com/orgs/nvidia/teams/riva/resources/riva_quickstart/files` (will need to log in to your ngc account).
3. From step 2, you will get CLI download command similar to this: **ngc registry resource download-version"nvidia/riva/riva_quickstart:2.8.1"**, type the command into the terminal of the Riva Server to download the Riva Quick Start Toolkit.



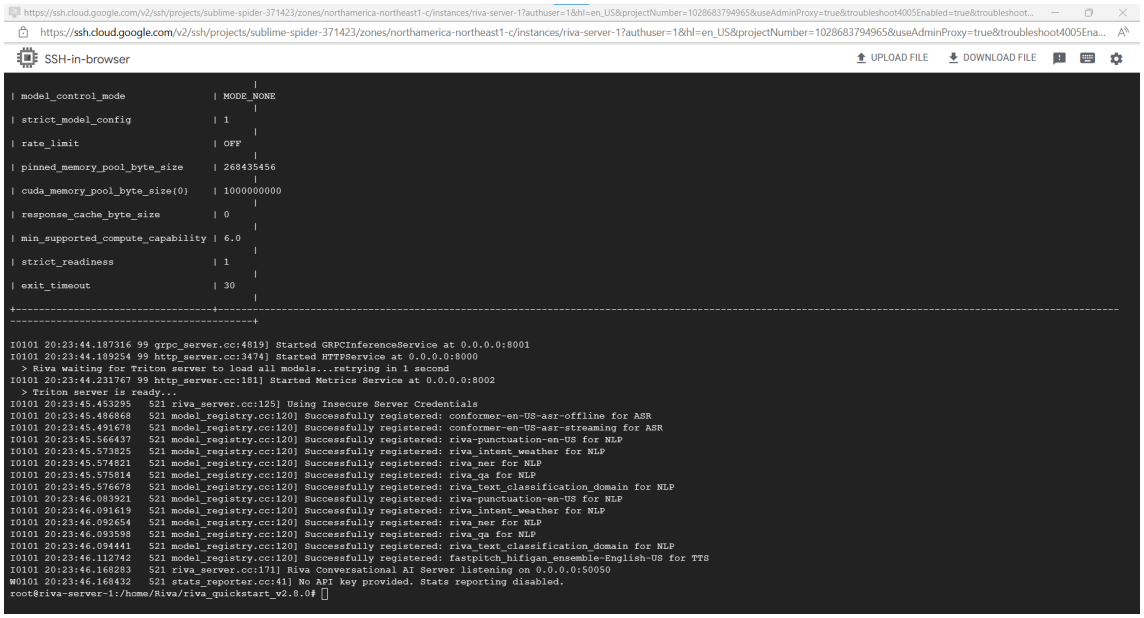
4. After the toolkit has been downloaded, run **riva_init.sh** to configure the Riva Docker environment.
5. After the Riva Docker environment has been configured, run **riva_start.sh** to start the Riva Server.



6. The Riva Server will take a few seconds to start, once it start, the Riva Server will be launched as a localhost at port 50050 as default. You can change the port number and relative config inside **config.sh**. (You will need to run **riva_init.sh** every time you changed the configuration inside of **config.sh**).

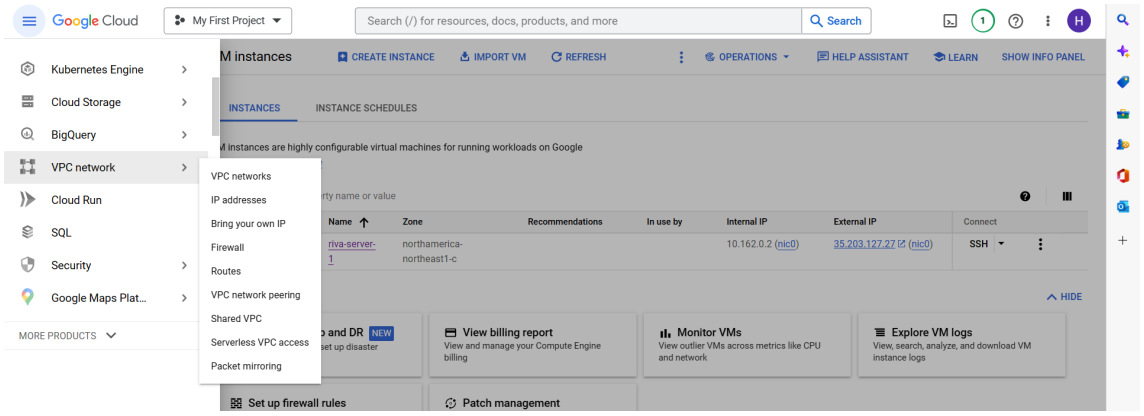


7. To get the log message of the running Riva Server, type: docker logs riva-speech.

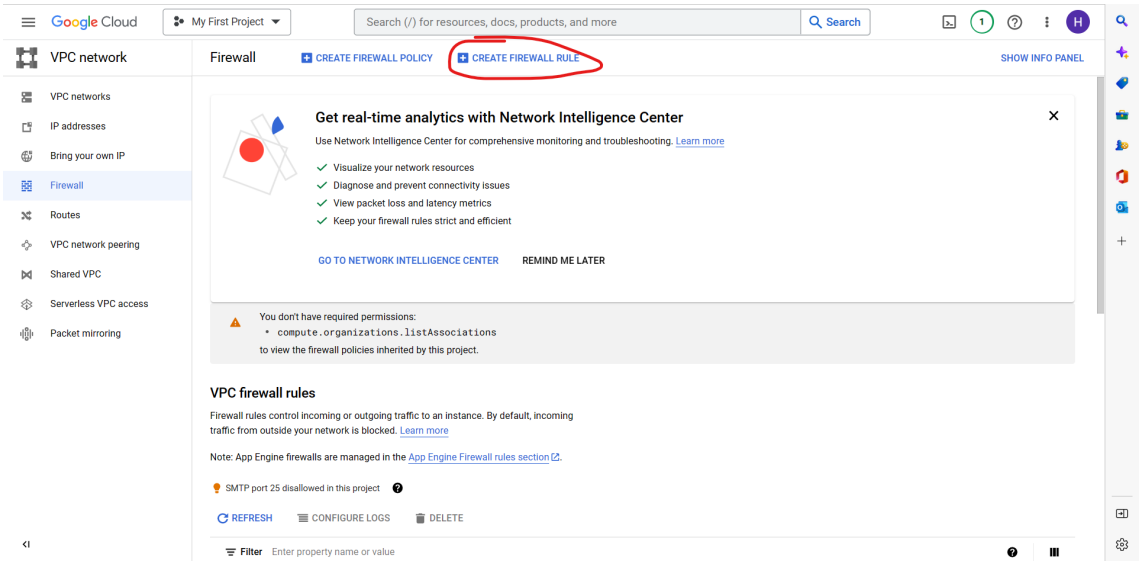


5.3 Access to the local host of the server

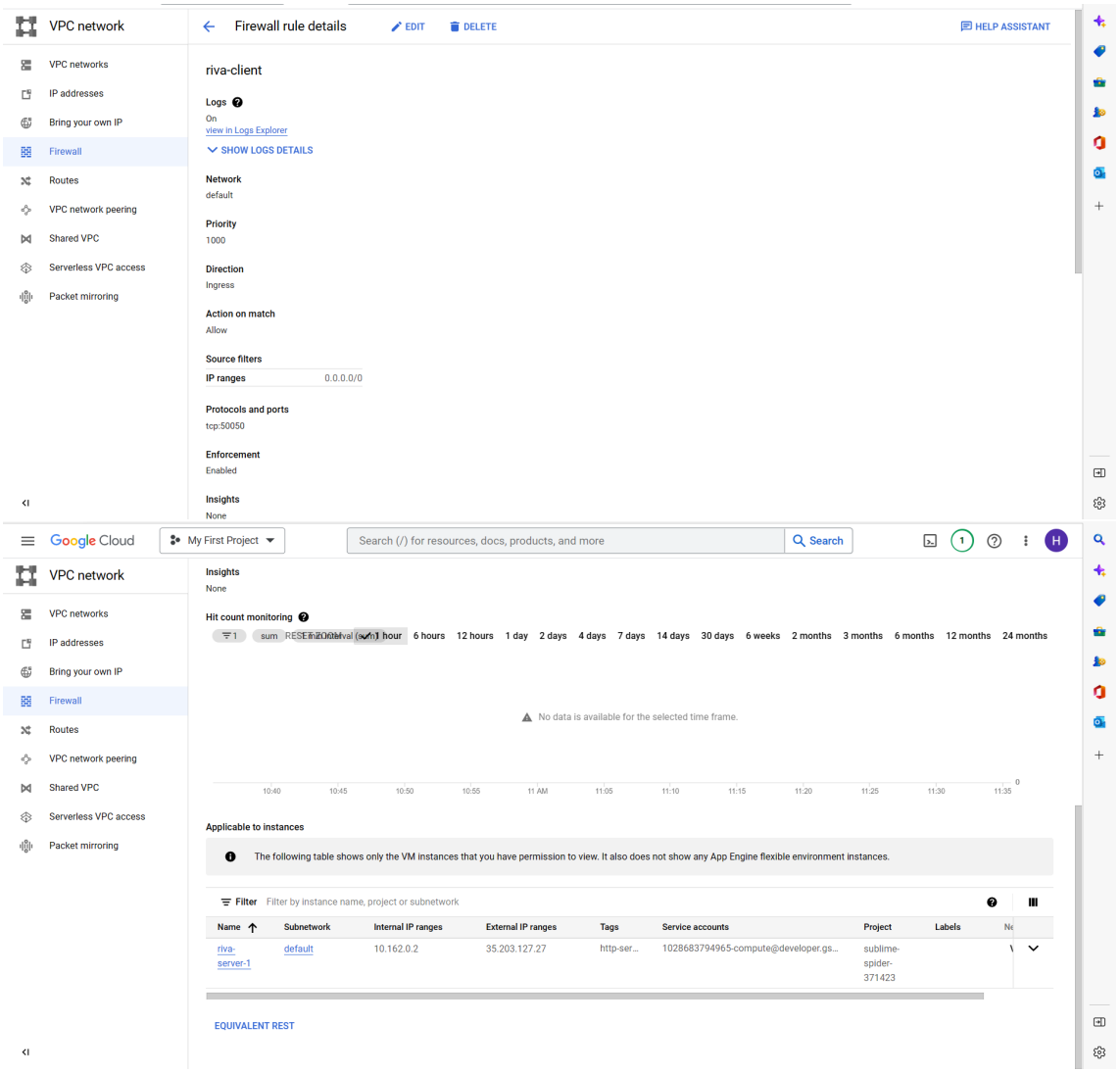
1. Make sure your Riva Server has already started.
2. Click on the *3-hyphen symbol*.
3. Click on **VPC network**.



4. Find the **Firewall** on the left hand side and click on it.
5. Find **CREATE FIREWALL RULE** and click on it.



6. We will set up a firewall rule, below is my setup.



7. After the firewall rule has been setup, you can access the local host server through the following syntax: external IP of your Server:Port of your Riva Server. For example, if the external IP of my Riva Server is: "12.34.56.789" and the port of my Riva Server is "50050", then the link to my Riva Server will be "12.34.56.789:50050". Notice that if you do not have a static ip address, the external IP address of your server will change everytime you restart your server.
8. The above link is useful to this python file: https://github.com/zslrmhb/Omniverse-Virtual-Assisstant/blob/main/config.py

6. Audio2Face

6.1 Background

We will be using Audio2Face to leverage the power of Riva Server and animate the virtual assistant.

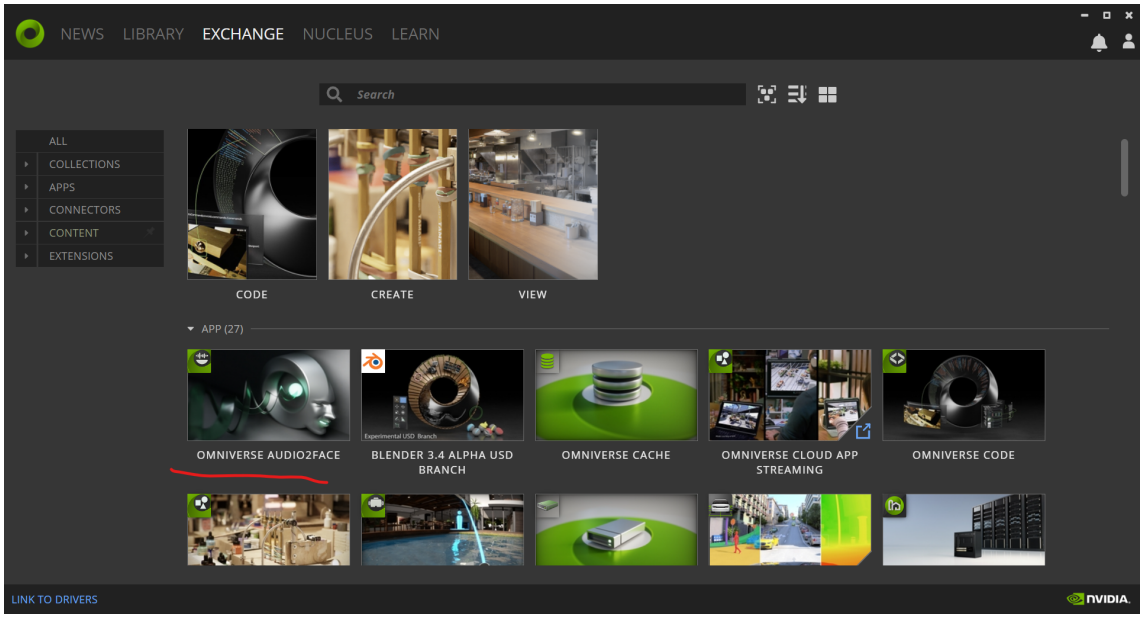
6.2 Audio2Face Setup

6.2.1 Download Audio2Face

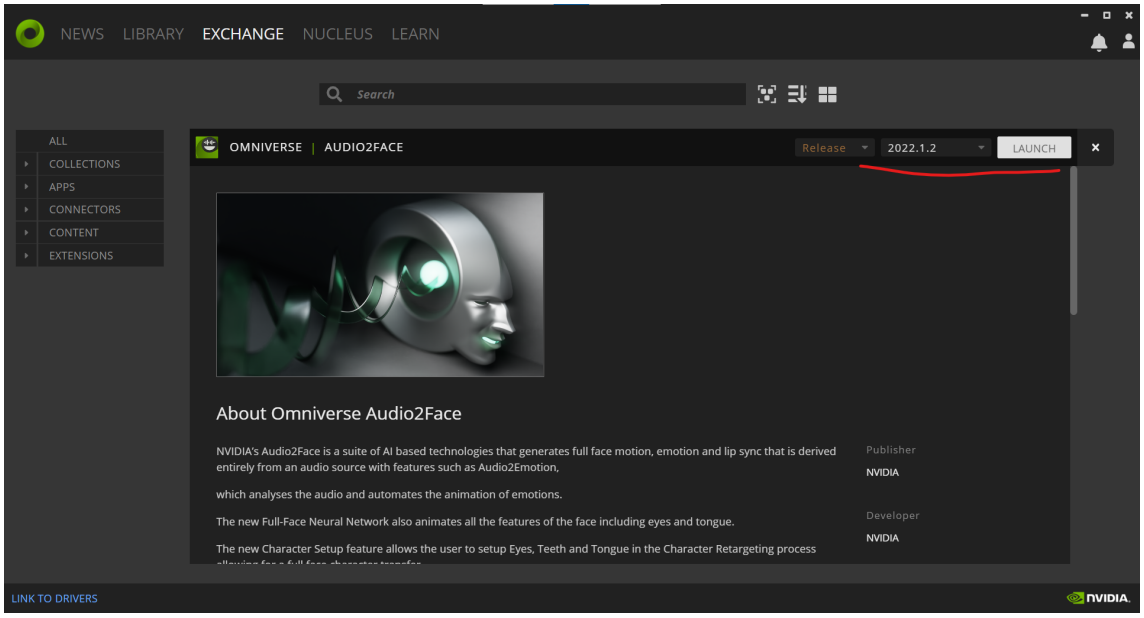
Prerequisites

- Windows 10, Version 1903 and Above (At the time of this documentation, only Windows is the supported platform.
- Have Nvidia Omniverse installed. If not, install through the following link: `https://www.nvidia.com/en-us/omniverse/download/`

1. Open the Omniverse Launcher.
2. Go to the **EXCHANGE** section.
3. Find **OMNIVERSE AUDIO2FACE**.

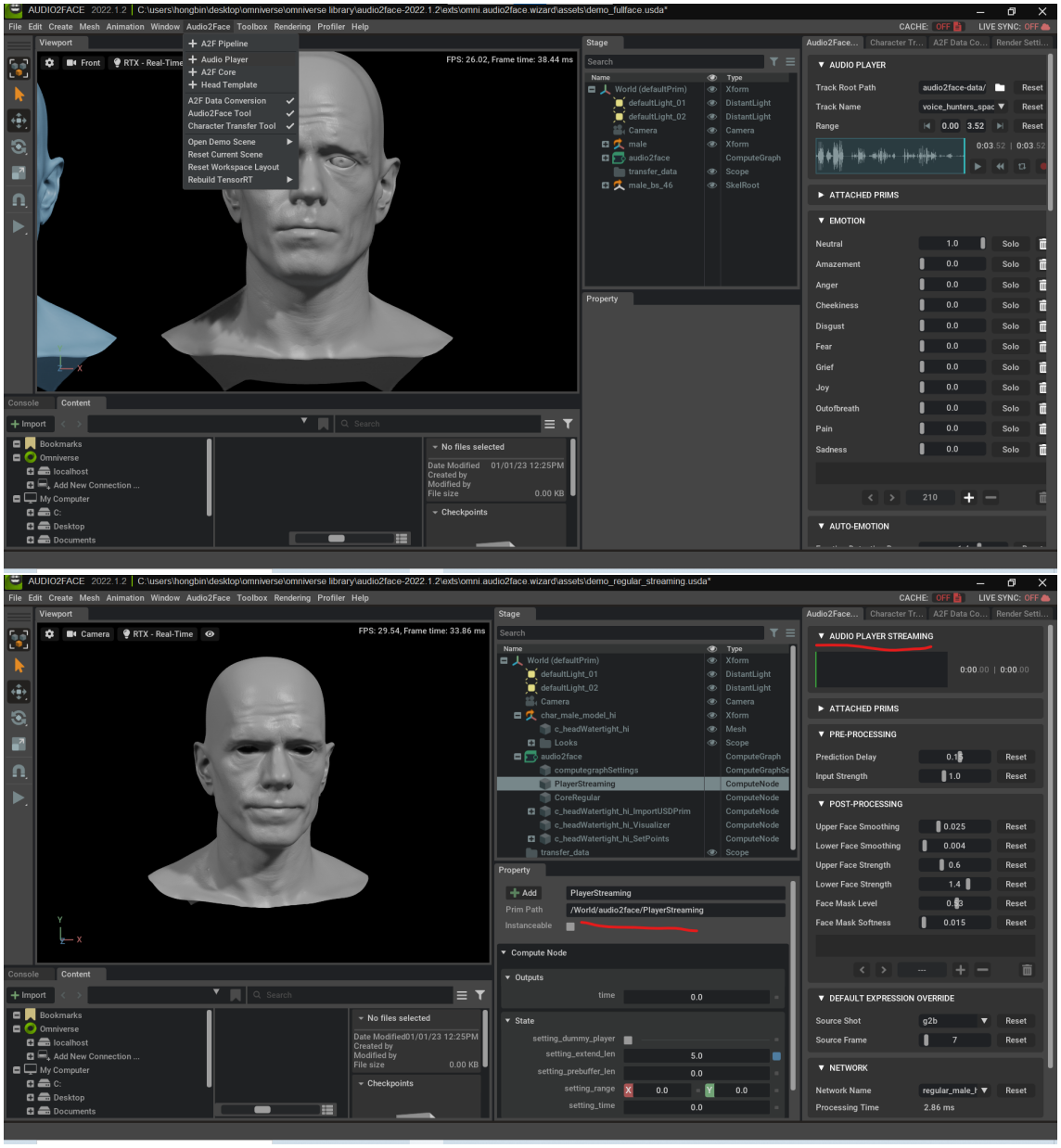


4. Select a version to download (I choose the latest one)



6.2.2 Integrate Audio2Face with Riva SDK

1. Launch the Riva Server.
2. Launch the Audio2Face and choose either the **Regular Core + Streaming Player** or the **Full Core + Streaming Player** Demo Scene.



3. Clone the following repository: `https://github.com/zslrmhb/Omniverse-Virtual-Assisstan git`
4. Go to the config.py of the cloned repo and type the URI in the following format: external IP address of your Riva Server:Port of yuor River Server. For example, if the external IP address is 12:34:56:789 and the port is 50050, then the URI in the config.py will be 12.34.56.789:50050.

Figure 6.1: External IP address example

5. Run main.py from the cloned repository.