**Lecture XX - Running Riva**

Notes:

Demonstrate how to install and run a Riva local server.

Connect to Jupyter Notebook and Omniverse Audio2Face.

# **Lecture Prerequisites**

1. You have access and are logged into **NVIDIA NGC**.
2. You have **Docker** installed with support for NVIDIA GPUs.
3. Docker Extension: **Volumes Backup & Share**
4. You have installed the **NGC CLI tool**.
5. **VScode**

**Warning:**

* Docker Containers for ML are very Large, and recommend to install all docker data into a Drive with at least 100 GB of free space.

# **Lecture Steps**

## **Step 1**

**Generate NGC API Key**

In the NGC website

Setup → API Key → Generate API Key

<https://ngc.nvidia.com/setup/api-key>

With the Generate NGC key we will:

1. Configure NGC CLI
2. Save it to the Environment Variables
3. Docker login into nvcr.io

**Configure NGC CLI**

Setup → CLI → Download

**Save Key to Environment Variables**

If your System is Windows:

**Using GUI**

* Using the windows search

**Edit the System Environment Variables**

* Using the Run Dialog box

Press Windows + R

Input: **SystemPropertiesAdvanced**

**Inside the window System Properties**

* **Click Environment Variables**

In System Variables click: **New**

* Inside the Variable Name and Variable Value add the following:

Variable Name: NGC\_API\_KEY

Variable Value: <API KEY obtained from the NGC Website>

Confirm Environment Variable on Powershell

In the PowerShell run the following command:

> $env:NGC\_API\_KEY

**Login into nvcr.io**

$ docker login nvcr.io

Username: $oauthtoken

Password: <Your Key>

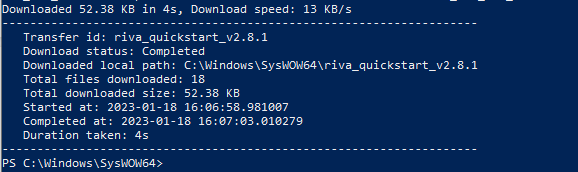
For the username, enter '$oauthtoken' exactly as shown.

azVsMWUxcjU2cWY5ODIwajl0YjR1bTE2cWk6YjJiMDQ2YjMtNWJlNy00YWY2LTgwOTUtZmI1ODY1NzFkMDI4

## **Step 2 - Setup NGC Registry**

**Download the scripts from the NGC Registry.**

$ngc registry resource download-version nvidia/riva/riva\_quickstart:2.8.1



Now enter the directory **riva\_quickstart\_v2.8.1**

Locate the riva\_quickstart\_v2.8.1 Directory.

$cd <whatever path>/riva\_quickstart\_v2.8.1

Usually the path is:

C:\Windows\SysWOW64\riva\_quickstart\_v2.8.1

Drive C:

Windows

SysWOW64

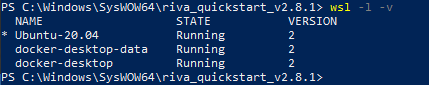
## **Step 3 - Initializing Riva**

In the Riva Quickstart Directory there are a series of useful Bash Scripts and the protos for the gRPC microservice.

Make sure we have Docker running and the correct WSL.

$wsl -l -v

Should return



$wsl -s Ubuntu-20.04

**Initialize Riva.**

The initialization step downloads and prepares Docker images and models.

**Star Riva**

The start script launches the server.

$bash riva\_init.sh

After Initialization is complete you should be able to start.

$bash riva\_start.sh

The Star Riva Script launches a Docker Container that contains all the models for RIVA (ASR, NLP, TTS) and tries to launch a Triton Server with gRPC at port 50051.

Recommend Launching the Docker Container Manually and create a Curated Volume with a few models given how compute intensive it can get.

**Download a Curated Docker Volume with the RIVA TTS models loaded.**

$ boredengineer2/riva-tutorial:riva-model-tts

Volume Name must be:

riva-model-tts

## **Step 4 - Creating a Docker Container**

Creating a Docker Container

We must create a RIVA TTS gRPC service at the localhost:50050 so we can later use it with Omniverse A2F.

$ docker run --init --gpus all -it -p 50050:50051 -v riva-model-tts:/data --name riva-speech-tts -p 8000:8000 -p 8001:8001 -p 8002:8002 nvcr.io/nvidia/riva/riva-speech:2.8.1

**Once completed we should be inside the shell of the running container.**

If not in the Shell environment we must run the following command:

$docker exec -it <container name> bash

If we already have a container we must start it.

$docker start <container name>

### **Docker Breakdown**

#### **Volume Mounting**

Docker Volumes:

$riva-model-repo (Created by the Riva Initialization Script).

$riva-model-tts (Curated custom Volume with only TTS models).

Must mount the volume named as /data directory inside the container, it is the standard format used by the Riva Triton Server pre-set Configurations.

-v riva-model-repo:/data

-v riva-model-tts:/data

#### **Docker Riva image path**

$ nvcr.io/nvidia/riva/riva-speech:2.8.1

#### **Docker Commands**

$docker run

--init

--gpus all

-it

--name riva-speech-tts

-v riva-model-tts:/data

-p 50050:50051

-p 8000:8000

-p 8001:8001

-p 8002:8002

## **Step 5 - Starting Riva a Service**

Starting Riva a Service.

### **Confirm if all the models are loaded**

Go to /data directory

$cd ../..

$ cd data/

$find models or $ ls

$ cd models

$ ls

Confirm if these models are loaded:

Models required for TTS

- fastpitch\_hifigan\_ensemble-English-US

- riva-onnx-fastpitch\_encoder-English-US

- riva-trt-hifigan-English-US

- spectrogram\_chunker-English-US

- tts\_postprocessor-English-US

- tts\_preprocessor-English-US

If everything is ok, return to the main directory

$ cd ../..

Inside the shell environment of the container we go to the bin directory.

$cd /opt/riva/bin

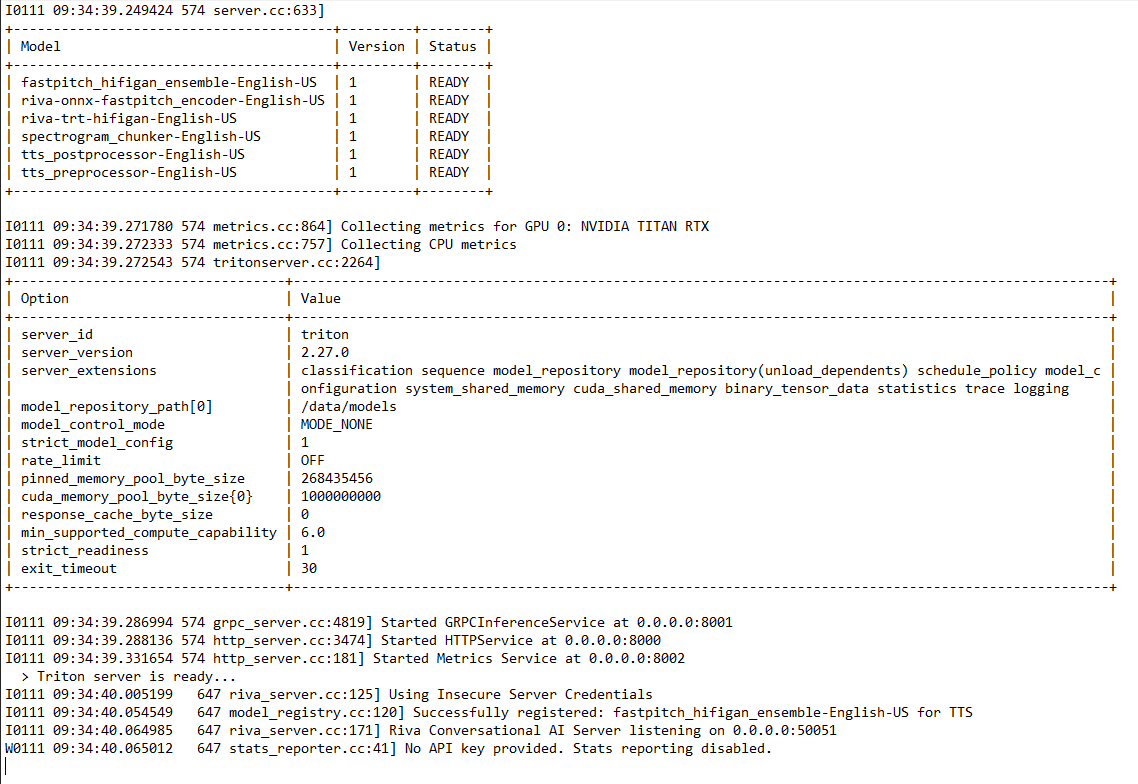
There will be 2 scripts (riva\_server and start-riva), we must run **start-riva**

$ ./start-riva

Monitor the shell output.

If Successful we must see:

> Triton server is ready…



## **Step 6 - Connecting to VSCode**

Connecting to the Riva Service with VScode.

**Working with Jupyter Notebooks on VSCode**

<https://code.visualstudio.com/docs/datascience/jupyter-notebooks>

Launch VSCode

Create a New folder where we will put the project

Create a Virtual Environment.

$python -m venv .venv

Install Jupyter

$pip install jupyter

Pull the Github repository with the Jupyter Notebook Examples.

\*\*My Curated repository\*\*

Nvidia has their Repository of Tutorials

<https://github.com/nvidia-riva/tutorials>

## **Step 7 - Connecting to Omniverse Audio2Face**

Connecting to Riva Service with Omniverse A2F.

Launch Omniverse

Launch A2F

# **Lecture Commands**

$ngc registry resource download-version nvidia/riva/riva\_quickstart:2.8.1

Locate the riva\_quickstart\_v2.8.1 Directory.

$cd <whatever path>/riva\_quickstart\_v2.8.1

$bash riva\_init.sh

$bash riva\_start.sh

# **Curating Docker Volume**

**Move the models directory**

$mv current\_directory target\_directory

$mv models models-all

**Make a new empty models directory**

$mkdir models

**Copy only the necessary model directory inside the models-all into the models directory.**

$cp -a current\_directory target\_directory

**example**

$cp -a /models-all/fastpitch\_hifigan\_ensemble-English-US models

$cp -a /models-all/riva-onnx-fastpitch\_encoder-English-US models

$cp -a /models-all/riva-trt-hifigan-English-US models

$cp -a /models-all/spectrogram\_chunker-English-US models

$cp -a /models-all/tts\_postprocessor-English-US models

$cp -a /models-all/tts\_preprocessor-English-US models

Generate NGC API key

Setup → API Key → Generate API Key

<https://ngc.nvidia.com/setup/api-key>

$ docker login nvcr.io

Username: $oauthtoken

For the username, enter '$oauthtoken' exactly as shown.

Now you can use Docker pull to get the image.

From the NGC website search for Riva Speech Skills container.

Once in the container look for Copy Image Path

Select the desired image

Then run the command to run the container

$ docker run --gpus all -it --rm -v local\_dir:container\_dir nvcr.io/nvidia/riva/riva-speech:x.x.x-server

Make sure to give the container an absolute path for local directory

The quickstart guide ask u to run the NGC CLI Tool

# **Riva Content**

Nvidia Riva Client Libraries Pypi

<https://pypi.org/project/nvidia-riva-client/>

Riva Client tutorials

<https://github.com/nvidia-riva/tutorials>

NGC Riva Speech Skills

<https://catalog.ngc.nvidia.com/orgs/nvidia/teams/riva/containers/riva-speech>

RIVA Docs

<https://docs.nvidia.com/deeplearning/riva/user-guide/docs/overview.html>

# **Docker Content**

Docker documentation about the bind mount and how to get a proper absolute path.

Start a container with a bind mount

<https://docs.docker.com/storage/bind-mounts/#start-a-container-with-a-bind-mount>

**Path conversion on Windows**

<https://docs.docker.com/desktop/troubleshoot/topics/>

**Docker - How to Move Docker to Another Drive on Windows**

<https://www.youtube.com/watch?v=gWBNk2KYg3M>

$ wsl -l -v

$ wsl --shutdown

$ wsl --export docker-desktop-data d:\docker-data\dockerdesktop.tar

$ wsl --unregister docker-desktop-data

$ wsl --import docker-desktop-data D:\docker-data\desktop D:\docker-data\dockerdesktop.tar