In this post I will go through the process of setting up NVIDIA Clara Train on AWS

I am referring to Brad Genereaux’s blog post to create a NVIDIA Clara based system.

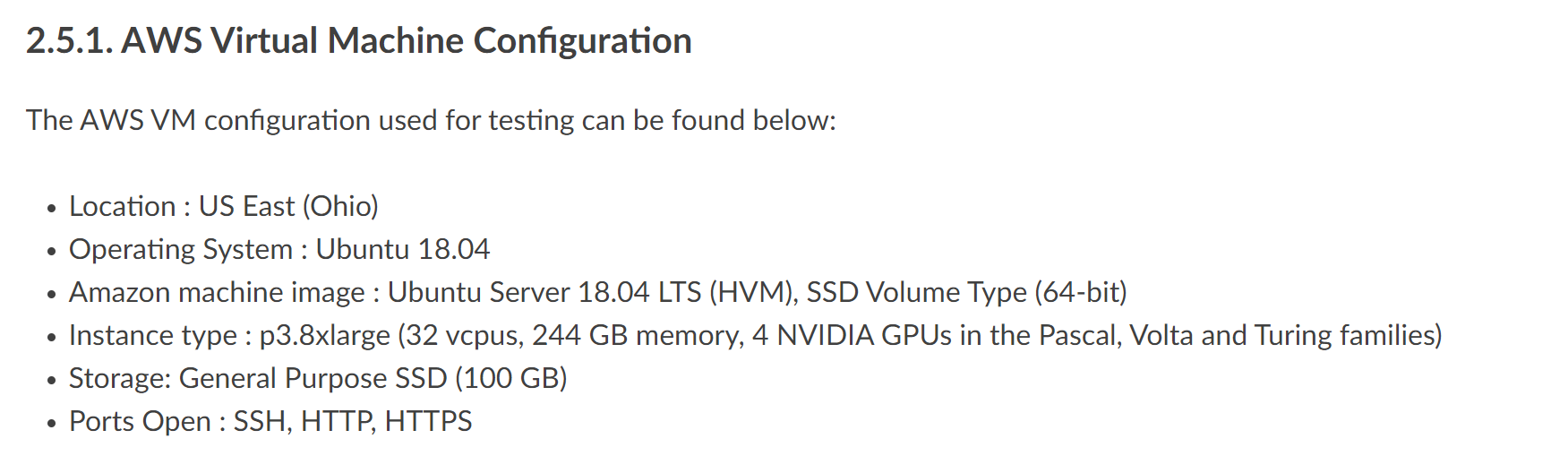
Brad’s blog - <https://medium.com/@integratorbrad/how-i-built-a-space-to-train-and-infer-on-medical-imaging-ai-models-part-1-24ec784edb62>

I will also use the NVIDIA Clara official installation guide and various other posts to install and troubleshoot.

**Disclaimer: The example shown here is NOT FOR CLINICAL USE and learning purpose only. The models are not FDA approved and not to be used for clinical decision making.**

**AWS Environment setup**

Create an AWS instance with following configuration (taken from NVIDIA official guide):

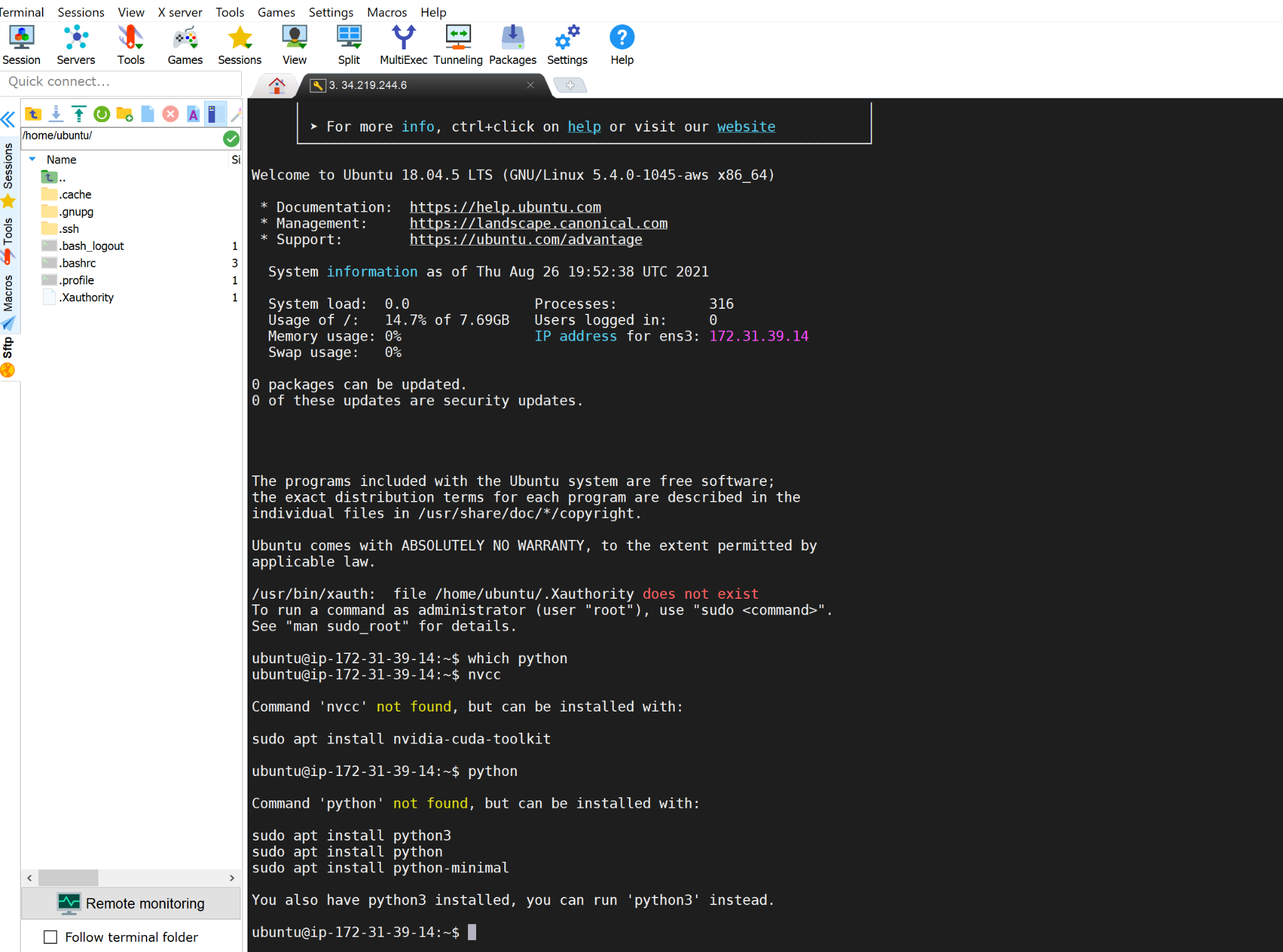


Ref: <https://docs.nvidia.com/clara/deploy/ClaraInstallation.html#installation-on-a-cloud-service-provider-csp>

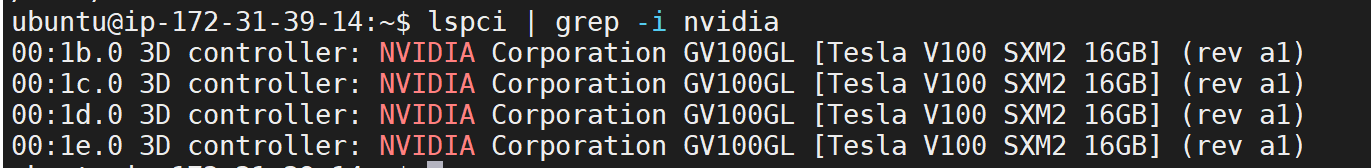
I will create a spot instance for my environment. P3.8xlarge is an expensive environment, by using a spot instance you can reduce the cost significantly, but it will create some interruption based on spot availability.

You might be ok with using some inexpensive GPU instances like g4dn, but for now I am going with NVIDIA suggested instance(p3)

After creating the spot instance, remote into the AWS server -

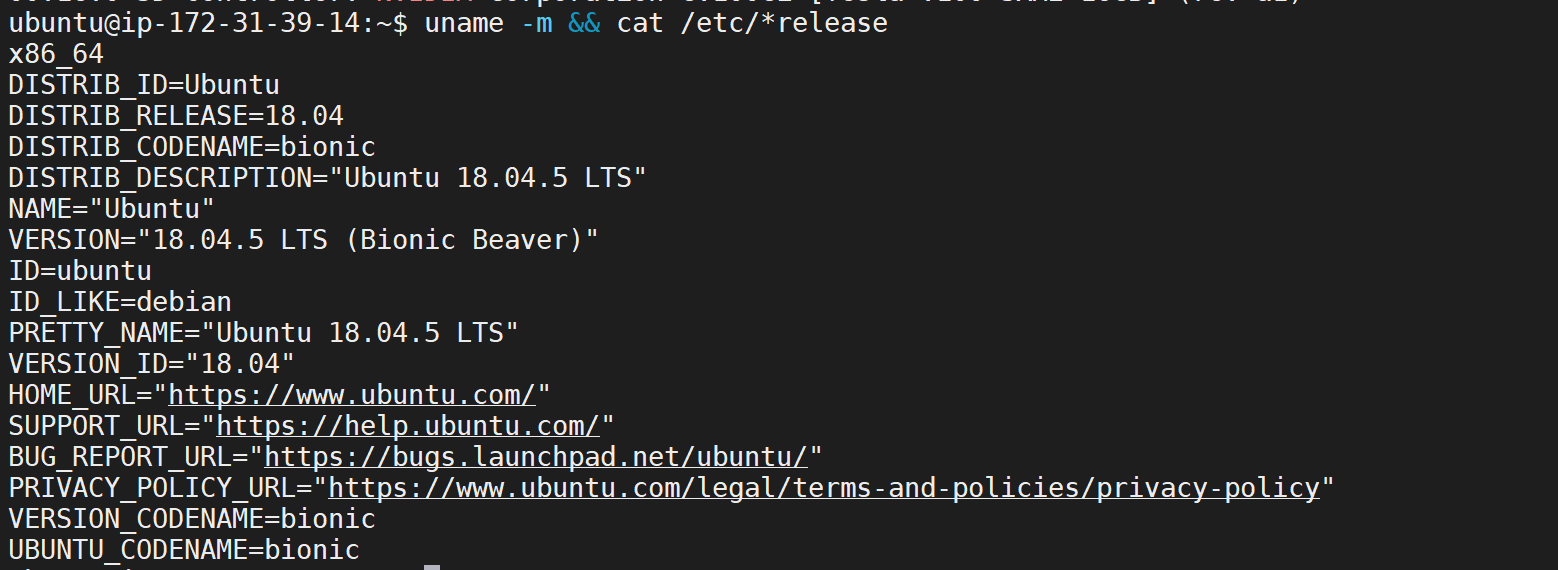


Check that you have a CUDA enabled GPU:



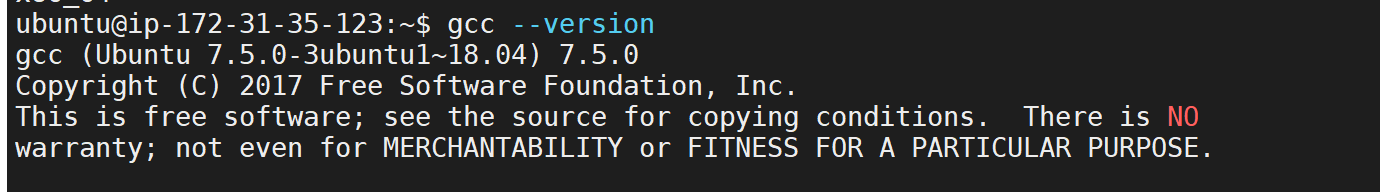
If nothing comes back from lspci command, then update the PCI hardware database of linux by entering `update-pciids` command and rerun the lspci | grep command.

Check for the CUDA supported version of linux:



It is a 64-bit system!

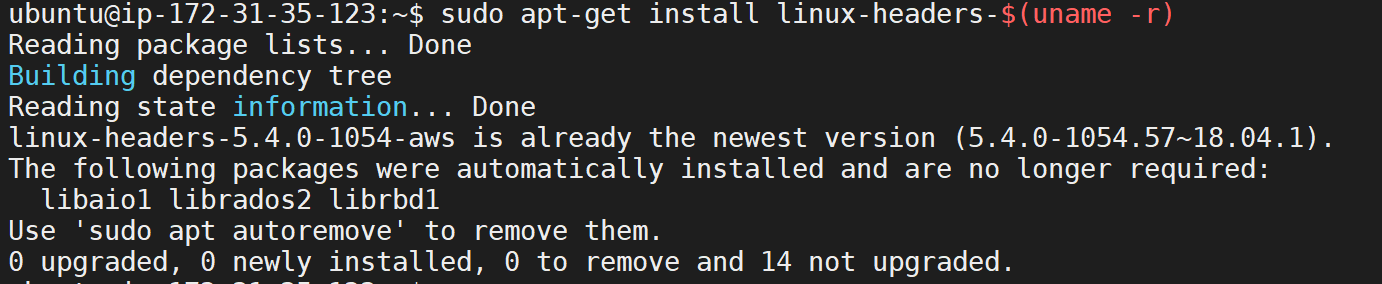
Verify that gcc is installed



Find out the kernel version of the system

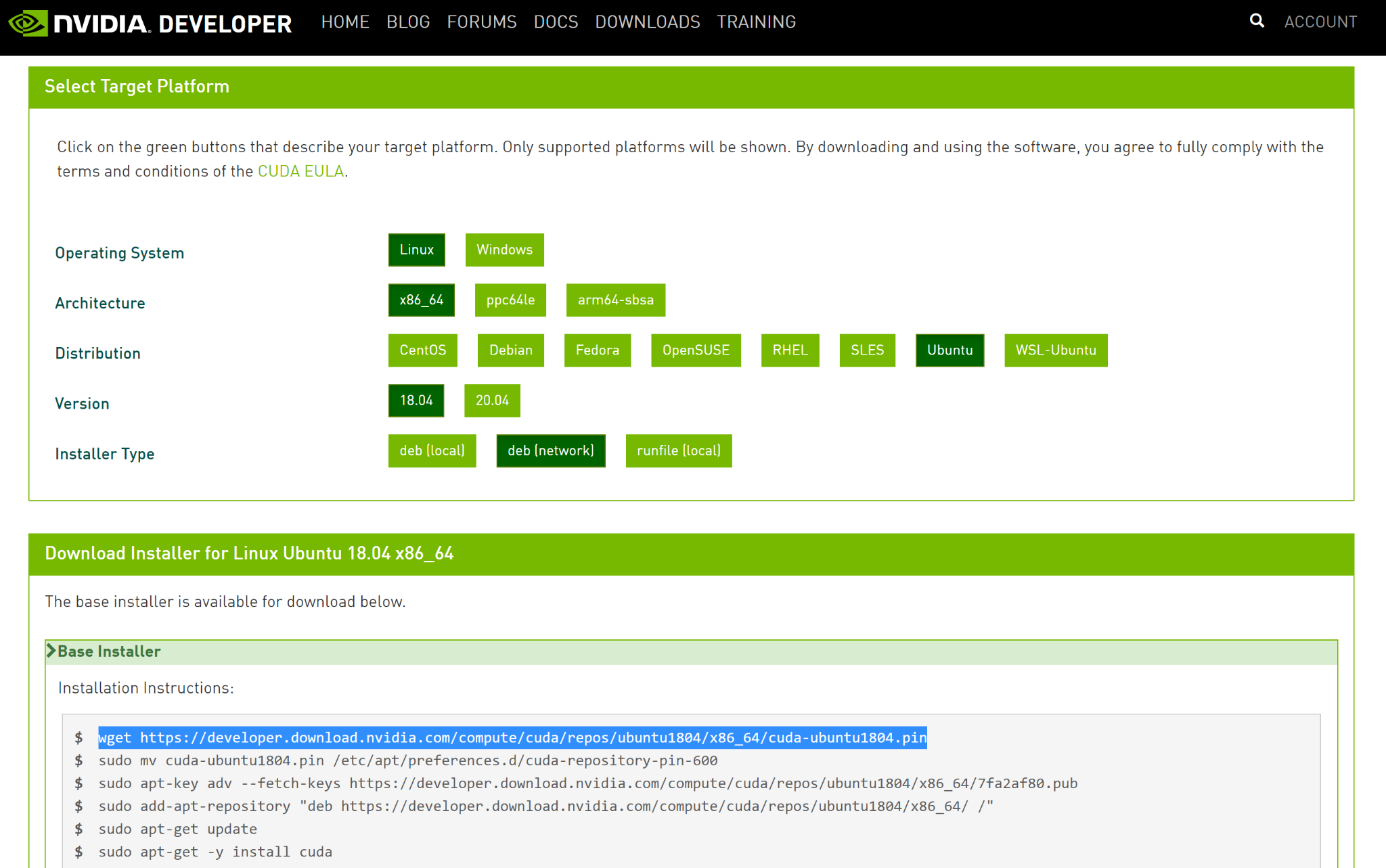


Before installing CUDA, the kernel header and development package of the same kernel version need to be installed.



Install CUDA by going to this link and selecting right choices:

<https://developer.nvidia.com/cuda-downloads?target_os=Linux>

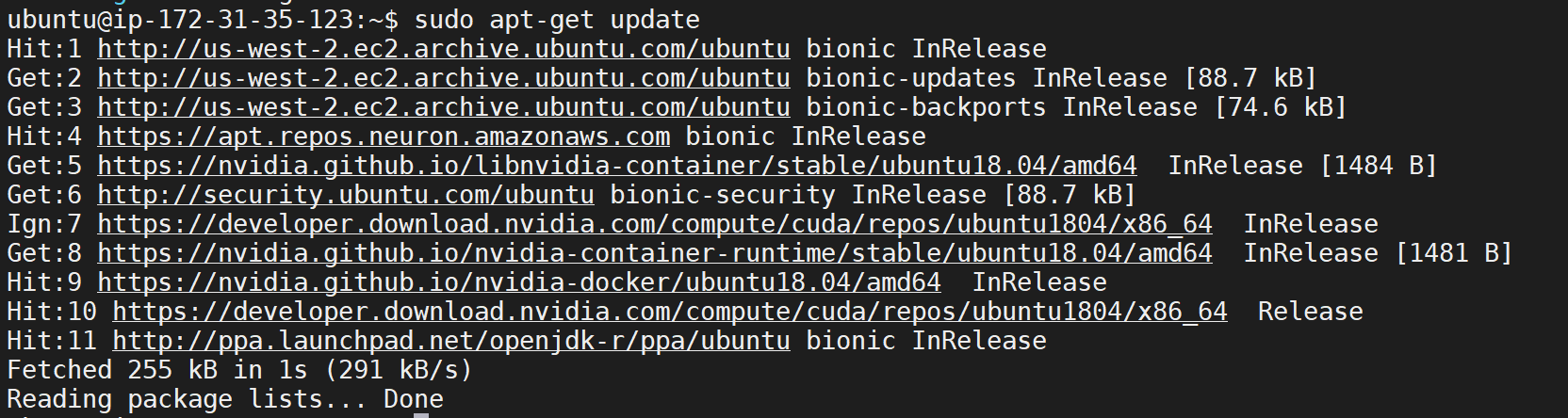


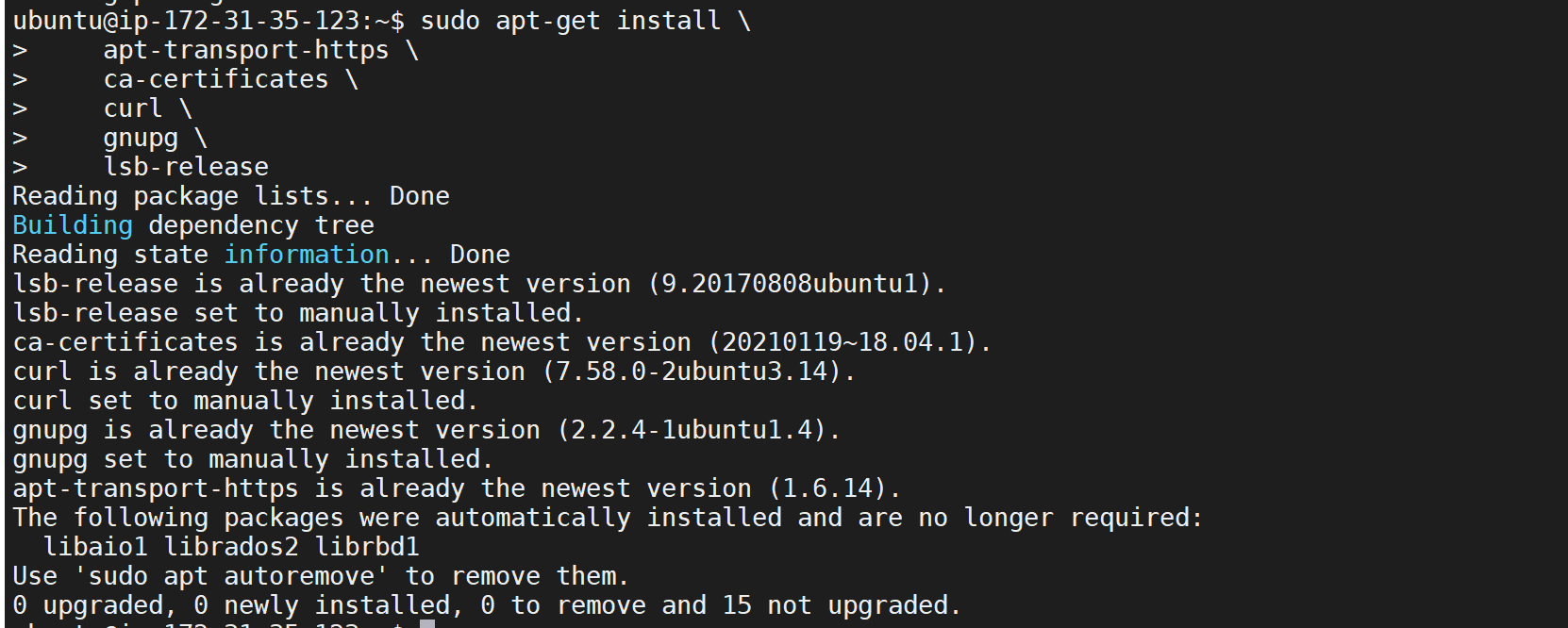
Reboot the system after you are done with the above steps



**Install Docker**

Follow the steps outlined in <https://docs.docker.com/engine/install/ubuntu/>





Add docker’s official GPG Key

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

Setup stable repository

echo \

"deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

Update apt package index

sudo apt-get update

Install docker community edition

sudo apt-get install docker-ce=5:19.03.8~3-0~ubuntu-bionic docker-ce-cli=5:19.03.8~3-0~ubuntu-bionic containerd.io

Verify that Docker is installed

sudo docker run hello-world

Add your user id in Docker user group

sudo usermod -aG docker $USER

Reboot

Sudo reboot now

**Install NVIDIA container toolkit**

Follow the steps outlined here -

<https://github.com/NVIDIA/nvidia-docker>

<https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/install-guide.html#docker>

Setup the stable repository and the GPG key-

distribution=$(. /etc/os-release;echo $ID$VERSION\_ID) \

&& curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo apt-key add - \

&& curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-docker.list | sudo tee /etc/apt/sources.list.d/nvidia-docker.list

After updating the package list install the nvidia-docker2

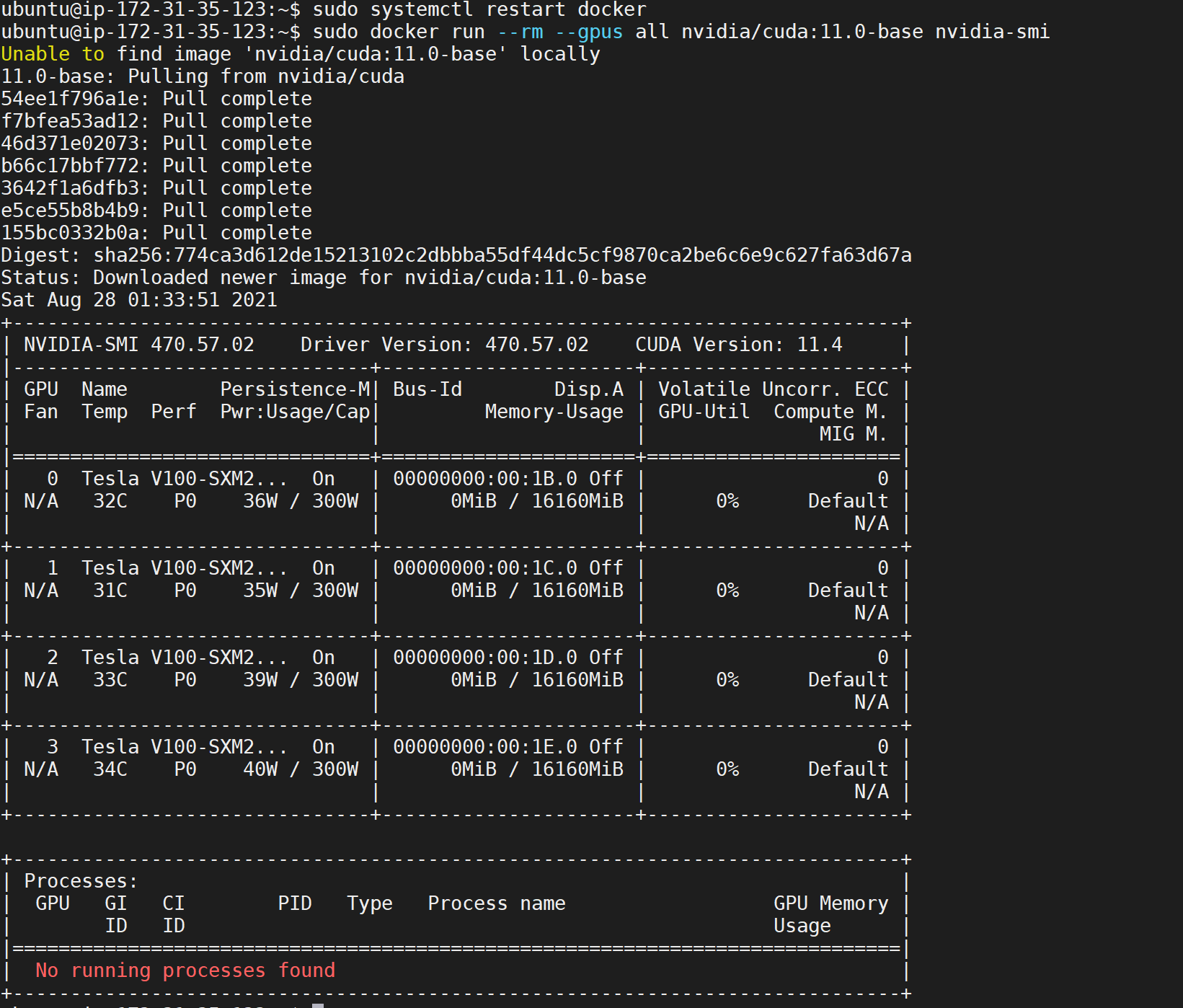
sudo apt-get update && sudo apt-get install -y nvidia-docker2

Restart docker demon

sudo systemctl restart docker

Test by running a base CUDA container

sudo docker run --rm --gpus all nvidia/cuda:11.0-base nvidia-smi



**Configuration of NGC access**

Login to NGC (<https://ngc.nvidia.com/>) and generate API Key and execute the following

mkdir /etc/clara/ngc

cd /etc/clara/ngc

wget<https://ngc.nvidia.com/downloads/ngccli_cat_linux.zip> && unzip ngccli\_cat\_linux.zip && rm ngccli\_cat\_linux.zip ngc.md5 && chmod u+x ngc

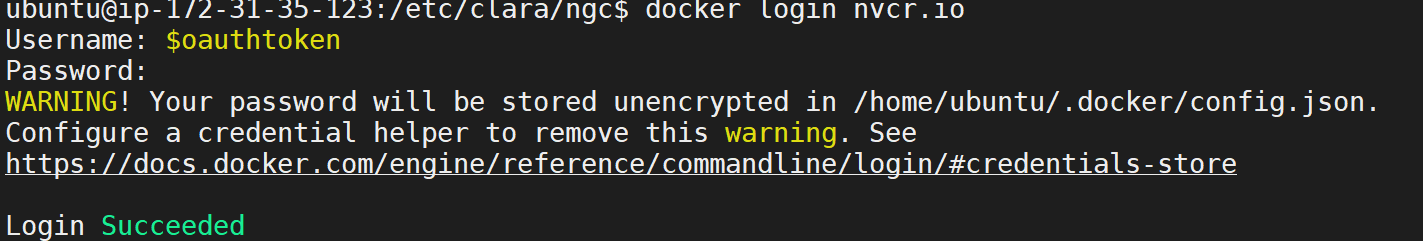
Add NGC key in ngc config

./ngc config set



Config docker to use NGC token

docker login nvcr.io



Now Ubuntu is loaded with Docker, NVIDIA docker, NVIDIA container toolkit

This picture shows the logical architecture of the Clara Train (taken from NVIDIA Clara github link given above).

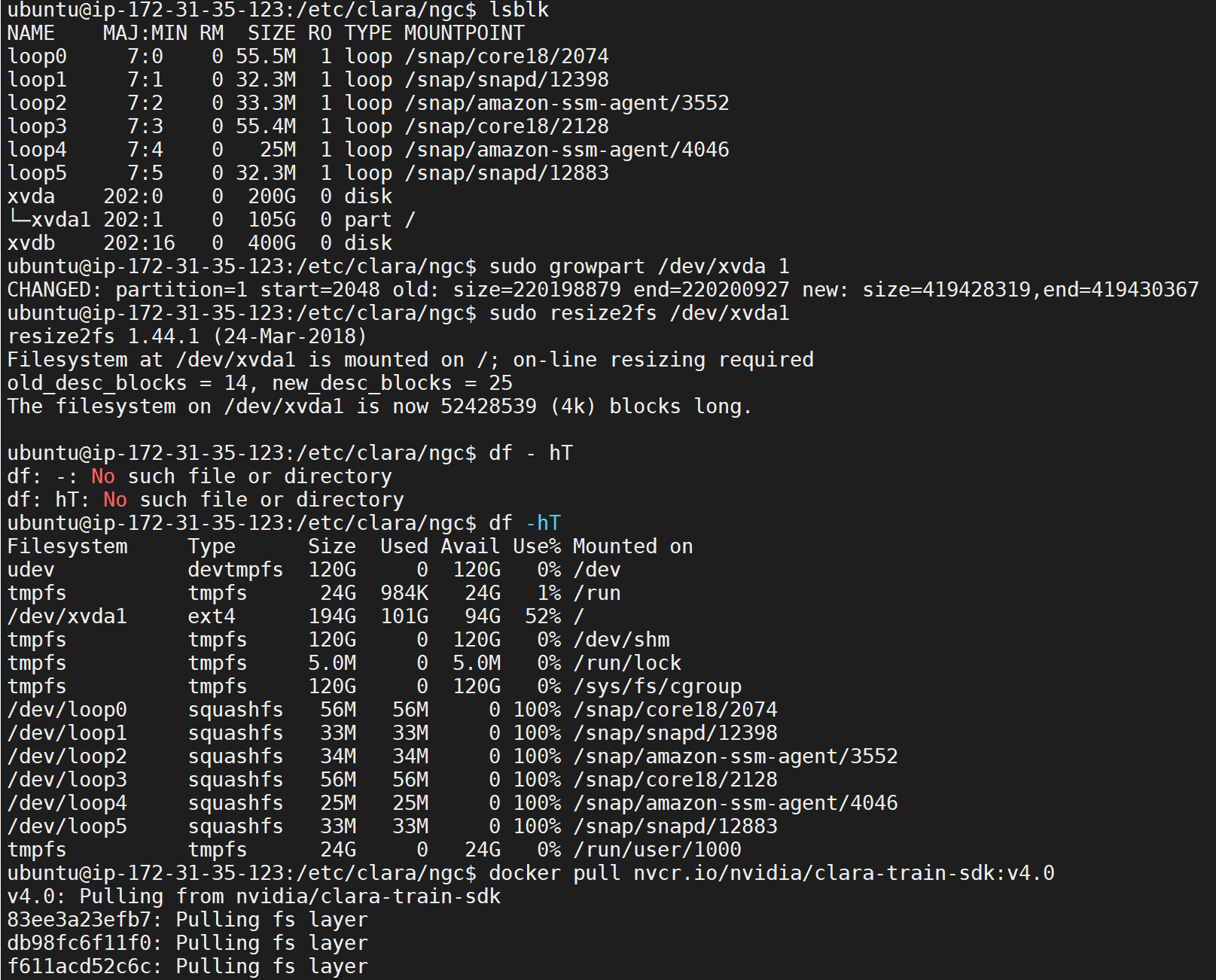


**Get the docker container for NVIDIA Clara Tarin SDK**

I am using the latest version available(v4)

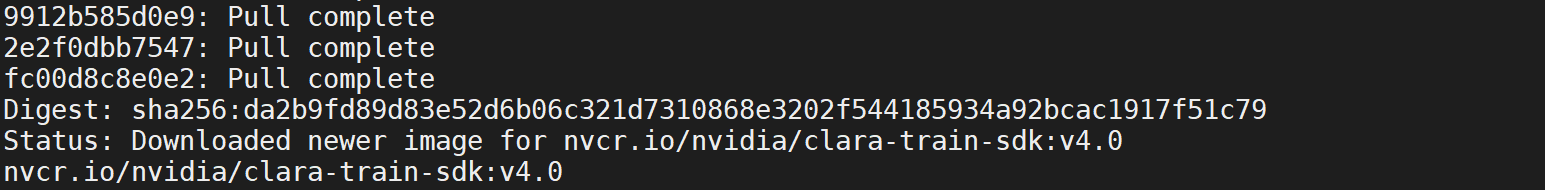
docker pull nvcr.io/nvidia/clara-train-sdk:v4.0

If you face problems with space, make sure to add and resize your drive.



Restart docker pull if the pull fails for any other reasons.

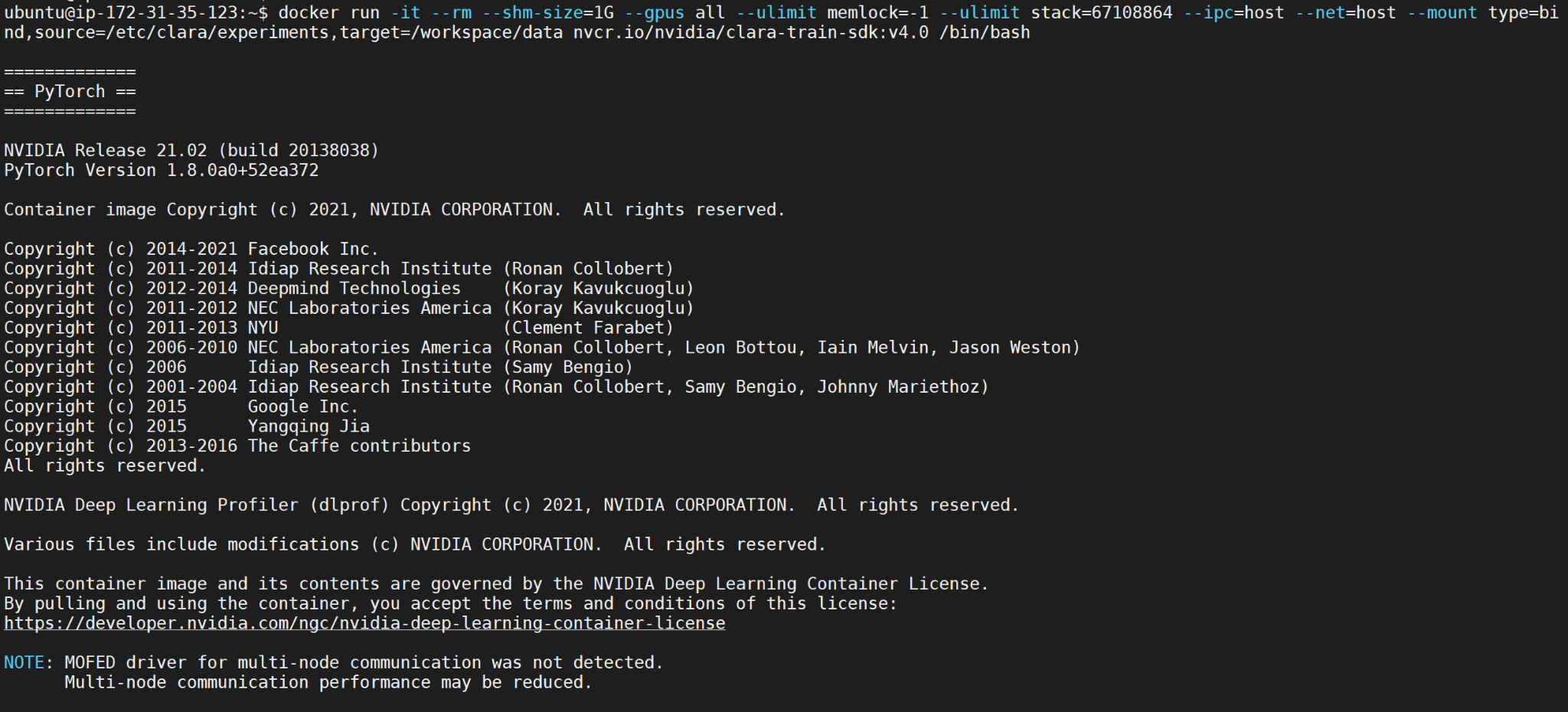
Successfully pulled clara train docker image:



Make a folder for experiments and change the ownership to user ubuntu:

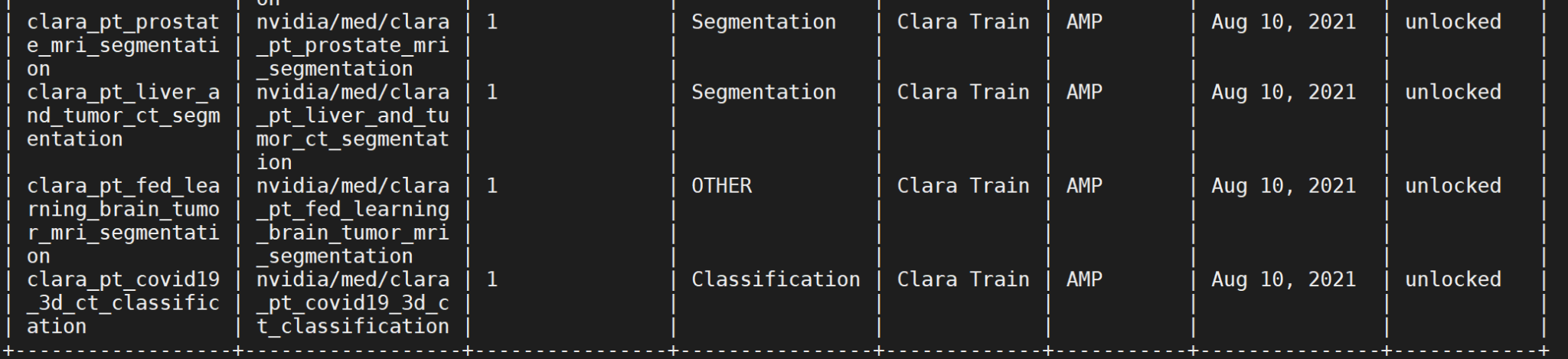
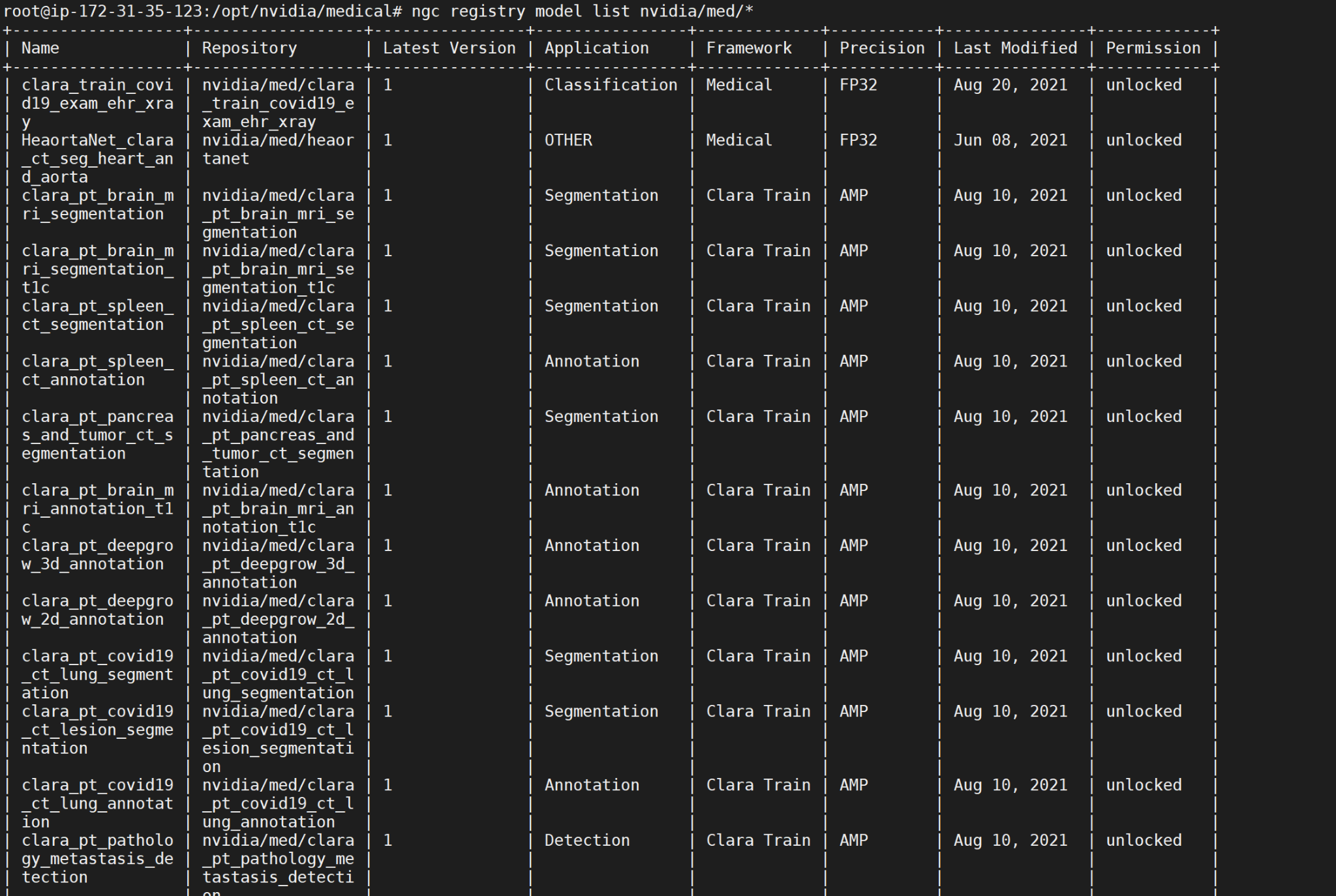


Go inside the clara train SDK by starting docker container in interactive mode:



Now you are inside the clara train docker container.

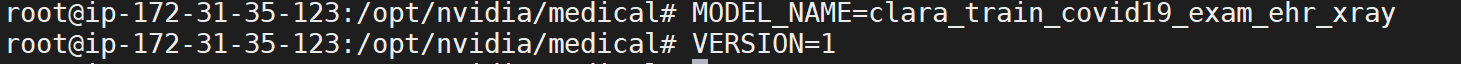
Run this command to get a full list of nvidia medical models.



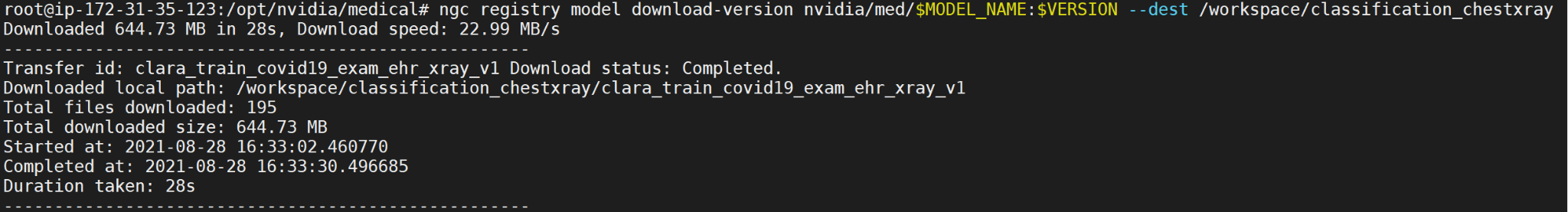
Create a folder for our 1st model - Chest xray



Set the parameters for the chosen model



Download the model.

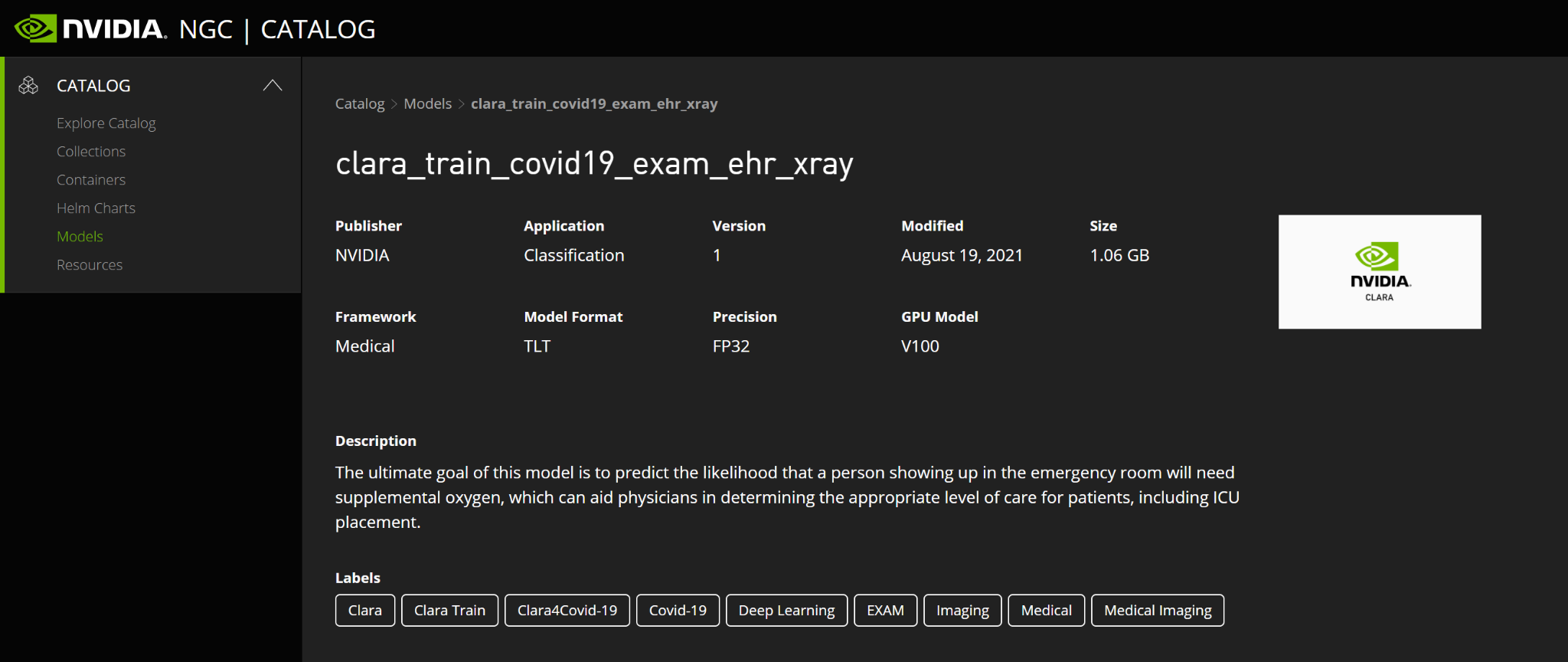


This will download the covid-19 chest xray classification model.

The details of the model available at <https://ngc.nvidia.com/catalog/models/nvidia:med:clara_train_covid19_exam_ehr_xray>

The description of the model as given in the above link: “Description

The ultimate goal of this model is to predict the likelihood that a person showing up in the emergency room will need supplemental oxygen, which can aid physicians in determining the appropriate level of care for patients, including ICU placement.”



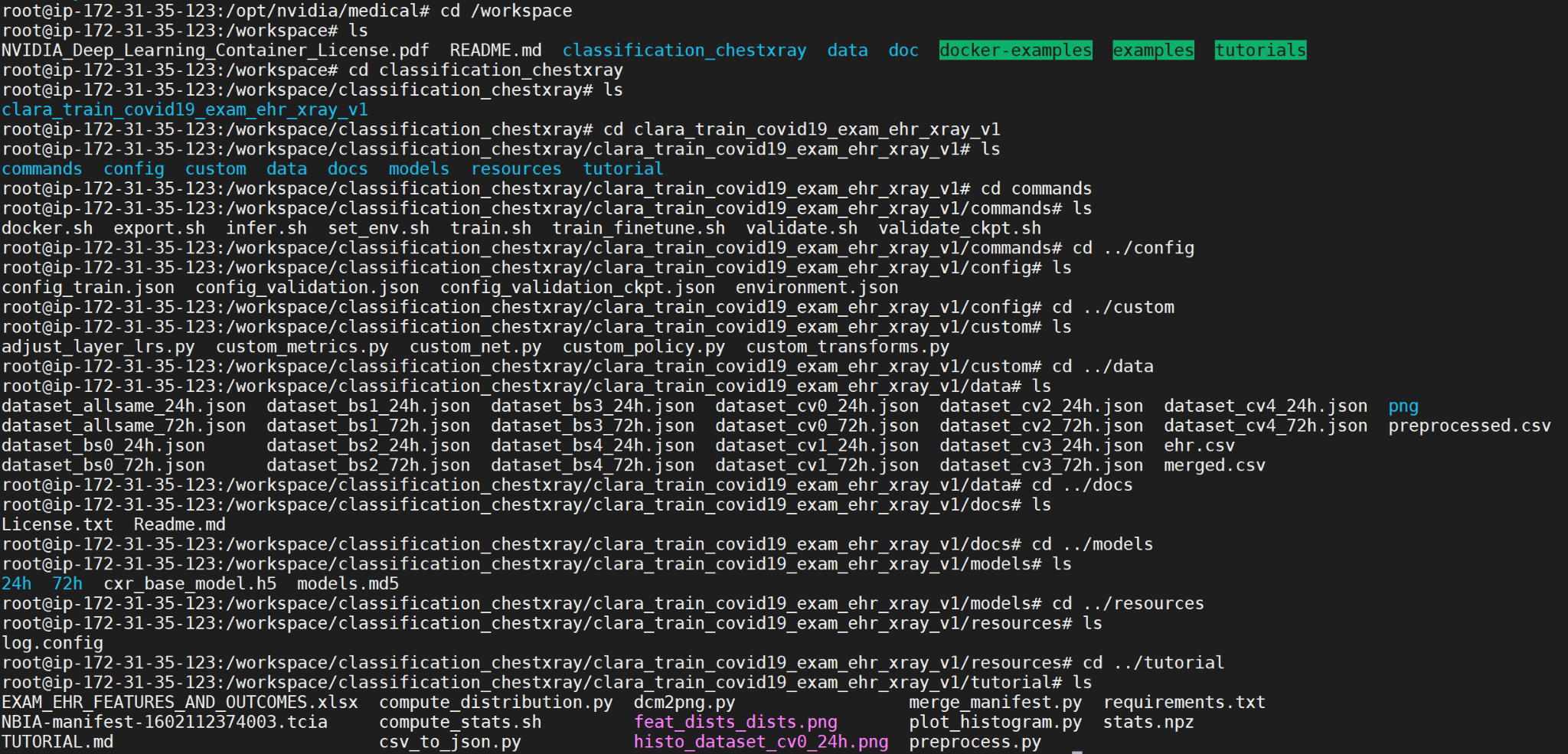
The model is in MMAR format. <https://docs.nvidia.com/clara/clara-train-sdk/pt/mmar.html>

This is how the model download directory looks like.



This has all the model weights, scripts and transforms.

Exploring the directory in a bit more details to see the contents



There you have it, Clara Train is up and running for use.