NVIDIA Clara AIAA

Dr Lorena Escudero
University of Cambridge

May 2020

AIAA Server

Amazon Web Services (AWS) Instance (EC2)

Step 1: Choose an Amazon Machine Image (AMI)

Cancel and Exit

Community AMIs (486)

GPU-Accelerated Cloud Gaming

More info

Categories

All Categories

Infrastructure Software (81)

DevOps (54)

Business Applications (5)

Machine Learning (18)

Industries (9)

Operating System

All Linux/Unix

Amazon Linux (43)

Ubuntu (35)

CentOS (1)

All Windows

Windows Server 2016 Base (7)

Windows Server 2019 Base (5)

NVIDIA Deep Learning AMI

Select

★★★★★ (2) | 20.03.1 Previous versions | By NVIDIA

Linux/Unix, Ubuntu Ubuntu 18.04 | 64-bit (x86) Amazon Machine Image (AMI) | Updated: 4/6/20

The NVIDIA Deep Learning AMI is an optimized environment for running the deep learning and HPC containers from the NVIDIA GPU Cloud (NGC) container registry. The deep learning containers on the NGC container registry require this AMI for GPU acceleration on AWS P3 and G4 GPU instances.

Product highlights:

- Provides AI researchers with fast and easy access to NVIDIA Volta and Turing GPUs in the cloud, with performance-engineered deep learning framework containers that are fully integrated, optimized, and certified by NVIDIA.
- Optimized for NVIDIA Volta and NVIDIA Turing GPU's for Highest Performance across a wide range of workloads
- NVIDIA accelerates innovation by eliminating the complex do-it-yourself task of building and optimizing a complete deep learning software stack tuned specifically for GPUs.

The NVIDIA Deep Learning AMI is an optimized environment for running the Deep Learning, Data Science, and HPC containers available from NVIDIA's NGC registry. The Docker containers available on the NGC container registry are tuned, tested, and certified by NVIDIA to take full advantage of NVIDIA Volta and Turing Tensor Cores, the driving force behind artificial intelligence. Deep Learning, Data Science, and HPC containers from the NGC registry require this AMI for the best GPU acceleration on AWS P3 and G4 instances. NVIDIA Deep Learning AMI Release Version 20.03.1 includes: Ubuntu Server 18.04 NVIDIA Driver 440.64.00 Docker-ce 19.03.6 NVIDIA Container Toolkit 1.0.5-1 Read more at: http://docs.nvidia.com/ngc/ngc-ami-release-notes/

NVIDIA Deep Learning AMI product detail page on AWS Marketplace

Show less

Step 2: Select instance type p3.2xlarge

NOTE: Your AWS account region needs to be US (N. Virginia) to be added to WU account, and also to find the NVIDIA Deep Learning AMI

AIAA Server

Security Group

Inbound rules | Outbound rules | Tags

Inbound rules

Edit inbound rules

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	0.0.0.0/0	-
SSH	TCP	22	::/0	-
Custom TCP	TCP	5000	0.0.0.0/0	-
HTTPS	TCP	443	0.0.0.0/0	-
Custom ICMP - IPv4	Echo Request	N/A	0.0.0.0/0	-
Custom ICMP - IPv4	Echo Request	N/A	::/0	-

ssh -i yourkey.pem ubuntu@ec2-XX-YYY-ZZZ-WWW.compute-1.amazonaws.com

```
Welcome to the NVIDIA GPU Cloud image.  This image provides an optimized
environment for running the deep learning and HPC containers from the
NVIDIA GPU Cloud Container Registry.  Many NGC containers are freely
available.  However, some NGC containers require that you log in with
a valid NGC API key in order to access them.  This is indicated by a
"pull access denied for xyz ..." or "Get xyz: unauthorized: ..." error
message from the daemon.

Documentation on using this image and accessing the NVIDIA GPU Cloud
Container Registry can be found at
  http://docs.nvidia.com/ngc/index.html
```

AIAA Server

Requirements

https://docs.nvidia.com/clara/tlt-mi/clara-train-sdk-v3.0/nvmidl/installation.html

1 - GPUs and NVIDIA drivers

```
ubuntu@ip-172-31-64-135:~$ nvidia-smi
Sat May 23 15:25:32 2020
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 440.64.00    Driver Version: 440.64.00    CUDA Version: 10.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-SXM2...  Off  | 00000000:00:1E.0 Off |                    0 |
| N/A   37C    P0    39W / 300W |      0MiB / 16160MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

2 - NVIDIA Container Toolkit

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

```
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

sudo apt-get update && sudo apt-get install -y nvidia-container-toolkit
sudo systemctl restart docker
```

AIAA Server

Get the Docker Container

Option A - available container (e.g. v3.0)

```
docker pull nvcr.io/nvidia/clara-train-sdk:v3.0
```

Option B - from .tar (pre-released) e.g. v3.1

1 - Get the file directly in AWS from Google Drive

```
function download-google(){
  echo "https://drive.google.com/uc?export=download&id=$1"
  mkdir -p .tmp
  curl -c .tmp/$1cookies "https://drive.google.com/uc?export=download&id=$1" > .tmp/$1intermezzo.html;
  code=$(egrep -o "confirm=(.+)&amp;id=" .tmp/$1intermezzo.html | cut -d"=" -f2 | cut -d"&" -f1)
  curl -L -b .tmp/$1cookies "https://drive.google.com/uc?export=download&confirm=$code&id=$1" > $2;
}
download-google 16XKmDWLL8WWnlgIHd3_2gvBh9I3IQkjC clara-ea31.tar.gz
```

2 - Load it to Docker

```
docker load -i clara-ea31.tar.gz

docker image load -i clara-ea31.tar.gz
```

NOTE: For this option, that involves having a local copy of .tar.gz of >8GB (>17GB uncompressed) and loading it to Docker, an initial disk space of >40 GB in the AWS instance is needed

AIAA Server

Run the container and start AIAA

1 - Run the Docker container

https://docs.nvidia.com/clara/tlt-mi/clara-train-sdk-v3.0/aiaa/installation.html

```
#export NVIDIA_RUNTIME="--runtime=nvidia -e NVIDIA_VISIBLE_DEVICES=1"
export NVIDIA_RUNTIME="--gpus all"
export OPTIONS="--shm-size=1g --ulimit memlock=-1 --ulimit stack=67108864”
export LOCAL_WORKSPACE=/var/lib/aiaa/
export REMOTE_WORKSPACE=/aiaa-experiments/
export LOCAL_PORT=5000
export REMOTE_PORT=80
export DOCKER_IMAGE="nvcr.io/nvidian/dlmed/clara-release:3.1.0"

docker run $NVIDIA_RUNTIME $OPTIONS -it -d --name aiaa-server --rm -p $LOCAL_PORT:$REMOTE_PORT -v
$LOCAL_WORKSPACE:$REMOTE_WORKSPACE $DOCKER_IMAGE start_aas.sh --workspace $REMOTE_WORKSPACE
```

2 - Inside Docker, start AIAA server with script start_aas.sh

```
export NVIDIA_RUNTIME="--gpus all"
export OPTIONS="--shm-size=1g --ulimit memlock=-1 --ulimit stack=67108864”
export LOCAL_WORKSPACE=/var/lib/aiaa/
export REMOTE_WORKSPACE=/aiaa-experiments/
export LOCAL_PORT=5000
export REMOTE_PORT=80
export DOCKER_IMAGE="nvcr.io/nvidian/dlmed/clara-train-sdk:v3.1-ga-qa-2"

docker run $NVIDIA_RUNTIME $OPTIONS -it -d --name aiaa-server --rm -p $LOCAL_PORT:$REMOTE_PORT -v $LOCAL_WORKSPACE:$REMOTE_WORKSPACE
```

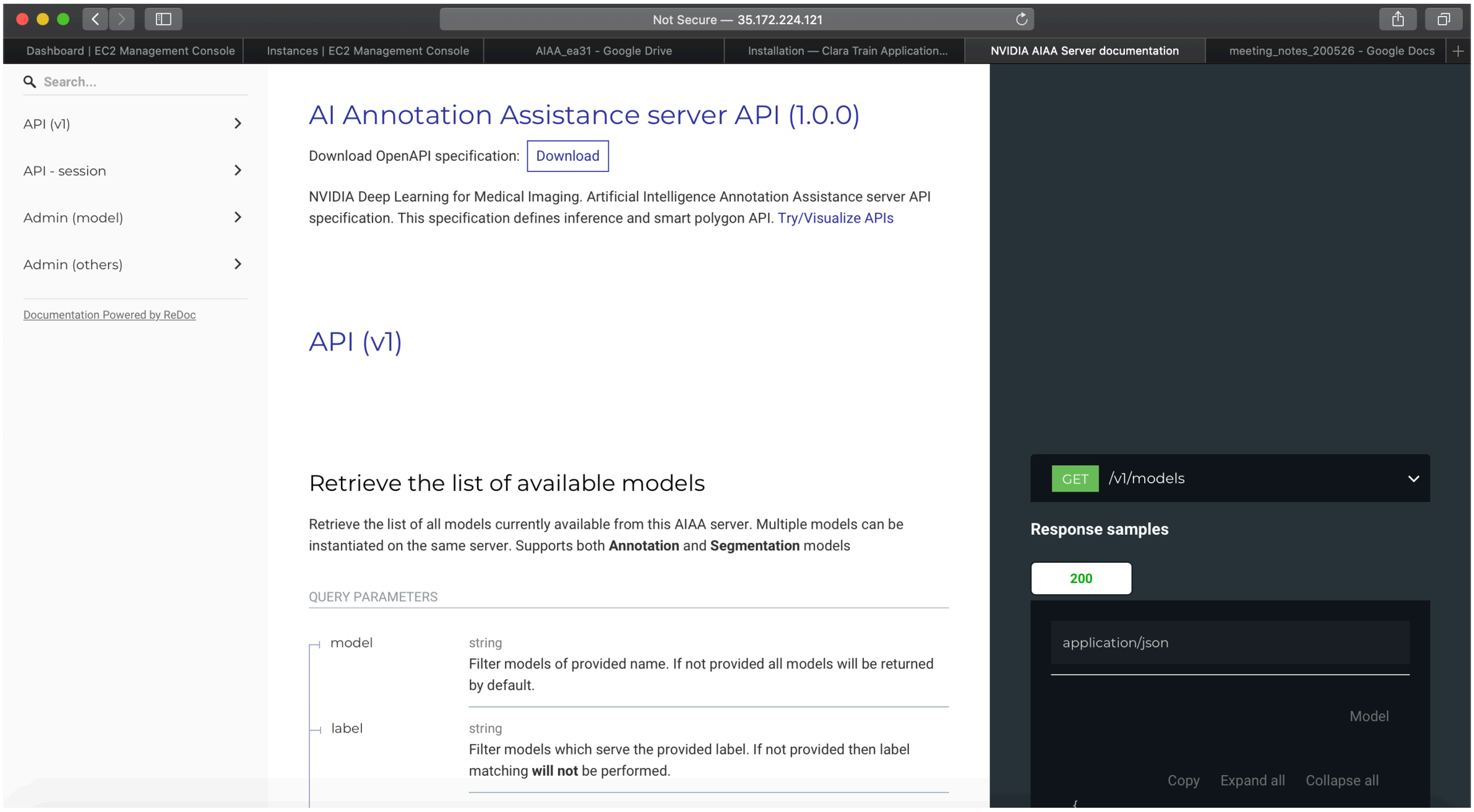
```
================
== TensorFlow ==
================

NVIDIA Release 20.03-tf1 (build 11025831)
TensorFlow Version 1.15.2

Container image Copyright (c) 2019, NVIDIA CORPORATION.  All rights reserved.
Copyright 2017-2019 The TensorFlow Authors.  All rights reserved.

NVIDIA Deep Learning Profiler (dlprof) Copyright (c) 2020, NVIDIA CORPORATION.  All rights reserved.

Various files include modifications (c) NVIDIA CORPORATION.  All rights reserved.
NVIDIA modifications are covered by the license terms that apply to the underlying project or file.

NOTE: MOFED driver for multi-node communication was not detected.
      Multi-node communication performance may be reduced.

root@b89a963ae105:/opt/nvidia# start_aas.sh --debug 1 &
```

AIAA Server

Check the AIAA server API

http://35.172.224.121:5000

(Note: this URL is only an example and will change with new instances)



AIAA Server

Upload the models

1 - Use curl PUT to add them

A - From NGC **https://docs.nvidia.com/clara/tlt-mi/clara-train-sdk-v2.0/aiaa/loading_models.html**

You need NGC CLI and a login (more info here)

1. create an NGC account and API token: https://ngc.nvidia.com/setup/api-key
2. Instructions on the right hand side for NGC CLI

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

unzip ngccli_cat_linux.zip
chmod +x ngc
md5sum -c ngc.md5

mv ngc /usr/local/bin

ngc config set
```

B - Locally:

```
#!/bin/bash

URL="0.0.0.0:5000"

# put clara seg spleen
model_name=clara_seg_spleen
curl -X PUT "http://$URL/admin/model/$model_name" -F "config=@$model_name/config/
config_aiaa.json" -F "data=@$model_name/models/model.trt.pb"
```

2 - Check they are available

curl http://35.172.224.121:5000/v1/models

[{"name": "clara_ann_spleen", "labels": ["spleen"], "description": "A pre-trained model for volumetric (3D) annotation of the spleen from CT image", "version": "3", "type": "annotation", "padding": 20, "roi": [128, 128, 128]}, {"name": "clara_deepgrow", "labels": [], "description": "2D DeepGrow model based on Unet", "version": "3", "type": "deepgrow"}, {"name": "clara_seg_spleen", "labels": ["spleen"], "description": "A pre-trained model for volumetric (3D) segmentation of the spleen from CT image", "version": "3", "type": "segmentation"}]%