



Singularity Image from Nvidia Clara V4.0 Docker Image

Anil Kemiseti; April 3, 2021

Nvidia Clara train is generally delivered as a docker container. This short how-to is part of the [📄](#) & [📄](#) series, which provides steps to create a singularity image from the Nvidia Clara docker image.

Step 1 | Getting Early Access for Clara Train SDK V4.0

Nvidia has opened up early access to Clara Train v4.0¹. The interest form is available here². The significant change for this version is moving away from tensorflow and adopting PyTorch. Also, this version is based on MONAI³. MONAI is an open-source framework to enable deep learning in healthcare imaging.

Step 2 | Saving The Docker Image Locally

Nvidia provides a docker image for Nvidia Clara sdk through Nvidia's NGC catalog⁴. Many research institutes prefer to use singularity containers on their HPC⁵ systems. To create this instance, I like to save⁶ the docker image locally as a first step. The command shown below is a Unix command. If you are running Windows operating system, I would advise using docker



desktop with the wsl2 backend. Singularity runs on Linux natively⁷ and can also be run on Windows and Mac through virtual machines (VMs).

```
$ docker save -o clara-train-sdkv4.0.img \
nvcv.io/ea-nvidia-clara-train/clara-train-sdk:v4.0-EA2
```

Figure 1: Command to save docker image locally

Step 3

Creating The Singularity Definition File

Use the singularity definition file⁸ shown in figure 2 is used to create the singularity image. Modify the path, which is shown in highlighted text. In this definition file, I have made the following changes.

1. Setting the PYTHONPATH environment variable. This helps in including custom code.
2. Install pip3, llvm, and clang. The latter was needed for installing sigpy python library.
3. Install nodejs and jupyterlab. These are needed for "jupyterlab-nvdashboard" extension.

This extension works only with jupyterlab version 1.0.0.

Save the above script as clara4.0EA2.def. Use the following command to build the singularity image. This command creates the image "clara-train-sdk-v4.0-ea2.sif."

```
$ sudo singularity build clara-train-sdk-v4.0-ea2.sif clara4.0EA2.def
```

Figure 3: Command to build singularity image

Step 4

Cloning The Nvidia Clara Example Git

This git⁹ provides examples for getting started with Clara. Clone this repo into a local directory. The Getting started jupyter notebook for v4.0 is present in the directory "PyTorch-Early-Access/NoteBooks"



Step 5

Adding Scripts To Launch JupyterLab

In the directory

"<directory-examples-cloned>/clara-train-examples/PyTorch-Early-Access/NoteBooks/scripts"
you will find scripts to install and launch the docker instance. In the same directory, place the following code in a file called "startSingularity.sh"

```
1  #!/bin/bash
2  SINGULARITY_IMAGE=/home/akemiseti/mydev/containers/clara-train-sdk-v4.0-ea2.sif
3  jupyterPort=$1
4  ##### check if parameters are empty
5  if [[ -z $jupyterPort ]]; then
6      jupyterPort=8890
7  fi
8  cmd2run="jupyter lab /claraDevDay --ip 0.0.0.0 --port ${jupyterPort} --allow-root --no-browser"
9  sudo singularity exec --nv -B ${PWD}/../:/claraDevDay/ --pwd ${PWD} ${SINGULARITY_IMAGE} ${cmd2run}
```

Figure 4: Singularity Jupyterlab Launch Script

Here I am aiming to launch only the jupyterlab, not AIAA. This script only takes the port as a parameter and uses all the GPUs. Change the path of the singularity image (highlighted text) in the script to your script's location.

Step 6

Adding Script To Launch Singularity Shell

Instead of launching the jupyterlab, if you want to just get the shell use the following script

```
1  #!/bin/bash
2  SINGULARITY_IMAGE=/home/akemiseti/mydev/containers/clara-train-sdk-v4.0-ea2.sif
3  jupyterPort=$1
4  ##### check if parameters are empty
5  if [[ -z $jupyterPort ]]; then
6      jupyterPort=8890
7  fi
8  cmd2run="/bin/bash"
9  sudo singularity exec --nv -B ${PWD}/../:/claraDevDay/ --pwd ${PWD} ${SINGULARITY_IMAGE} ${cmd2run}
```

Figure 5: Singularity Shell Launch Script

Brief Look at Getting Started Notebook

As you can observe in step 5 and step 6, the parent directory where the scripts are located is bound to "/claraDevDay". When you launch the script, you would be able to access the following jupyterlab.

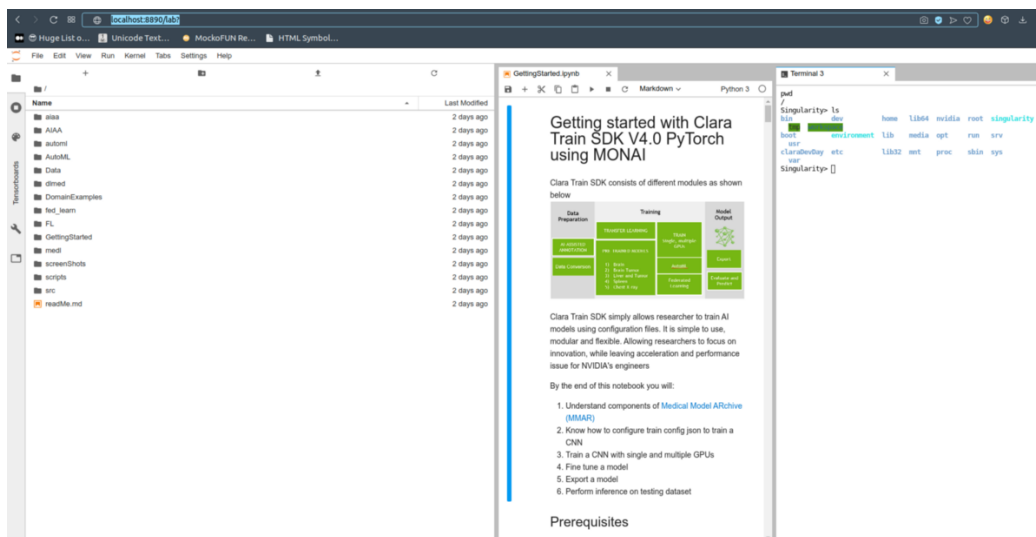


Figure 6: A look at Nvidia Clara Examples



```
1 Bootstrap: docker-archive
2 From: /home/akemiseti/mydev/containers/clara-train-sdkv4.0.img
3 %environment
4 PYTHONPATH=/home/akemiseti/mydev/fastMRI:$PYTHONPATH
5 export PYTHONPATH
6 %post
7 # python section
8 wget -O - https://apt.llvm.org/llvm-snapshot.gpg.key | apt-key add -
9 apt-get update
10 apt-get install -y python3-pip
11 apt install -y lsb-release wget software-properties-common
12 apt-get install -y clang-10 lldb-10 lld-10
13 apt-get install -y libllvm-10-ocaml-dev libllvm10 \
14 llvm-10 llvm-10-dev llvm-10-doc llvm-10-examples llvm-10-runtime
15 ln -s /usr/bin/llvm-config-10 /usr/bin/llvm-config
16 pip3 install pytest
17 pip3 install pylint
18 pip3 install nose
19 pip3 install pytest-mpl
20 pip3 install pixiedust
21 pip3 install scikit-learn
22 pip3 install pillow
23 pip3 install jupyterlab-nvdashboard
24 pip3 install sigpy
25 pip3 install ipympl
26 pip3 install ipywidgets
27 pip3 install itkwidgets
28 echo -----
29 echo -- fix bokeh issue downgrad to 1.4.0
30 pip3 uninstall -y bokeh
31 pip3 install bokeh==1.4.0
32 echo ----- bokeh installed
33 # install nodejs
34 curl -sL https://deb.nodesource.com/setup_12.x | bash -
35 apt-get update && apt-get install -y build-essential nodejs
36 # fixing python version back to python3
37 unlink /usr/bin/python
38 ln -s /usr/bin/python3 /usr/bin/python
39 python --version
40 echo -----
41 nodejs --version
42 which node
43 pip3 install jupyterlab==1.0.0
44 python -c "from shutil import which; print(which('node'))"
45 python -c "from jupyterlab.jlpmapp import which; print(which('node'))"
46 which npm
47 npm --version
48 echo -----
49 jupyter labextension install jupyterlab-nvdashboard
50 echo -----
51 echo -----jupyterlab intallation completed
52 jupyter lab build
53 echo -----
54 echo "Container build done!"
```

Figure 2: Singularity Definition File



Links

- [1] NVIDIA Clara Imaging. <https://developer.nvidia.com/clara-medical-imaging>, September 2018.
- [2] NVIDIA Clara Train 4.0 Interest Form. <https://developer.nvidia.com/nvidia-clara-train-40-interest-form>, November 2020.
- [3] Project MONAI. <https://monai.io/>.
- [4] Catalog | NVIDIA NGC. <https://tinyurl.com/nvidia-ngc>.
- [5] What is high performance computing?
- [6] Docker save. <https://docs.docker.com/engine/reference/commandline/save/>, April 2021.
- [7] Installation — Singularity container 3.0 documentation. <https://sylabs.io/guides/3.0/user-guide/installation.html>.
- [8] Definition Files — Singularity container 3.0 documentation. https://sylabs.io/guides/3.0/user-guide/definition_files.html.
- [9] NVIDIA/clara-train-examples. NVIDIA Corporation, February 2021.