Exxact Machine Learning Images (EMLI) - Quick Start Guide

Congratulations on purchasing an EMLI System. A Docker integrated system for Deep Learning.

Docker images available on this system:

- nvidia/cuda
- nvidia/caffe
- nvidia/digits
- nvidia/mxnet
- nvidia/theano
- nvidia/cntk
- partners/chainer
- portainer
- tensorflow/tensorflow:latest-gpu
- PyTorch
- RapidsAI

Docker Command line Option:

To pull additional Docker image (from NGC Repository)

```
# Download / pull images for NGC Repository

root@u105724:~# docker pull nvcr.io/nvidia/cuda:9.1-devel
9.1-devel: Pulling from nvidia/cuda
976a760c94fc: Already exists
c58992f3c37b: Already exists
0ca0e5e7f12e: Already exists
f2a274cc00ca: Already exists
708a53113e13: Already exists
2ec2fca7a49c: Pull complete
34026c3e50ea: Pull complete
0e4a761cbcd3: Pull complete
2d1d54944b4e: Pull complete
Digest: sha256:5c91a161147220b06624cc490877b5b3867c13e86d5ee40d0e0fe6d5117f2137
Status: Downloaded newer image for nvcr.io/nvidia/cuda:9.1-devel
nvcr.io/nvidia/cuda:9.1-devel
root@u105724:~#
```

View pulled images on system

```
# Run docker images command to see installed images

[root@localhost emli]# docker images
REPOSITORY                          TAG                        IMAGE ID       CREATED       SIZE
portainer/portainer                 latest                     10383f5b5720   2 weeks ago   78.6MB
nvcr.io/nvidia/rapidsai/rapidsai    cuda10.1-runtime-centos7   01b6c34e3c63   4 weeks ago   8.68GB
nvcr.io/nvidia/tensorflow           20.01-tf1-py3              e9f1a32f9cad   2 months ago  8.39GB
nvcr.io/nvidia/digits               20.01-tensorflow-py3       1430fdae6f40   2 months ago  9.36GB
nvcr.io/nvidia/tensorflow           20.01-tf2-py3              8fe085738892   2 months ago  7.15GB
nvcr.io/nvidia/mxnet                20.01-py3                  4c60607811c0   2 months ago  6.11GB
nvcr.io/nvidia/pytorch              20.01-py3                  5c0c8c90f238   2 months ago  9.12GB
nvcr.io/nvidia/caffe                20.01-py3                  6094e9a70920   2 months ago  4.85GB
nvcr.io/nvidia/cuda                 10.2-devel-centos7         f30f507196a1   3 months ago  2.83GB
nvcr.io/nvidia/cuda                 10.2-runtime-centos7       f545d1487da8   3 months ago  1.38GB
nvcr.io/nvidia/caffe2               18.08-py3                  e82334d03b18  19 months ago  3.02GB
nvcr.io/nvidia/theano               18.08                      1462ba2d70fe  19 months ago   3.7GB
nvcr.io/nvidia/cntk                 18.08-py3                  f92a52188dba  19 months ago  6.17GB
nvcr.io/partners/chainer            4.0.0b1                    4f3dd7135093   2 years ago   2.02GB
```

View all containers on the system (including running and stopped)

```
# docker ps // to see all active containers/container info
# docker ps -a // to see all containers/container info

[root@c101086 ~]# docker ps -a
CONTAINER ID   IMAGE                 COMMAND             CREATED              STATUS
PORTS                                NAMES
6a4d72fc7197   nvidia/digits         "python -m digits"  30 seconds ago       Up 29
seconds      0.0.0.0:5000->5000/tcp, 6006/tcp  digits-333201001-0
cdae95d22e84   portainer/portainer  "/portainer"        About a minute ago   Up About
a minute  0.0.0.0:9000->9000/tcp           portainer-021237417-0
```

Run command inside of the container (interactively)

```
# to execute a shell within the container
[root@c101086 ~]# docker run --gpus all --rm -it nvidia/cuda bash
root@6c83ee4f8141:/#
# you will see hostname change to the container ID you are now in
```

For additional docker images, please go to: https://hub.docker.com/

NVIDIA Digits

DIGITS Quickstart Script (found in the root's home folder Directory and /usr/local/bin)

This is now also loaded in /usr/local/bin/startDigits so you may run #startDigits from anywhere to start a new unique container

```
[root@localhost ~]# cat startDigits.sh
#!/bin/bash
DATE=$( date +%N )

docker run --gpus all -it --name digits-$DATE-0 -d -p 5000:5000 -v
/data/datasets:/opt/datasets --restart=always nvcr.io/nvidia/digits:20.01-tensorflow-
py3

# Using /data/datasets on the host for Digits to access the data files
#options
# --runtime=nvidia, specific for passing the nvidia-docker
# -e NVIDIA_VISIBLE_DEVICES="0,1,2,3" control which Nvidia GPU to pass to the
container
# --name = to name the container of your container
# -$DATE-0 variable implemented to create unique container names when starting a new
one
# -d = detached process to run the container in the background
# -p = specify port (host port:container port)
# -v = volume, to link a directory from the host system to the container (host
directory: container directory)
# --restart=always, set container to start after every restart
# nvidia/digits = specified docker image to load container
# in summary, this script will create a nvidia/digits based container using GPUs
0,1,2,3.  Container name = digits-$date-0 listening at port 5000 and linking the
/home/data host filesystem to /opt/datasets within the container.
# you can access the web GUI via web browser // use <hostsystemIP>:5000
```

Portainer

Portainer is a simple management solution for Docker. Easily manage your Docker hosts and Docker Swarm clusters via Portainer web user interface.

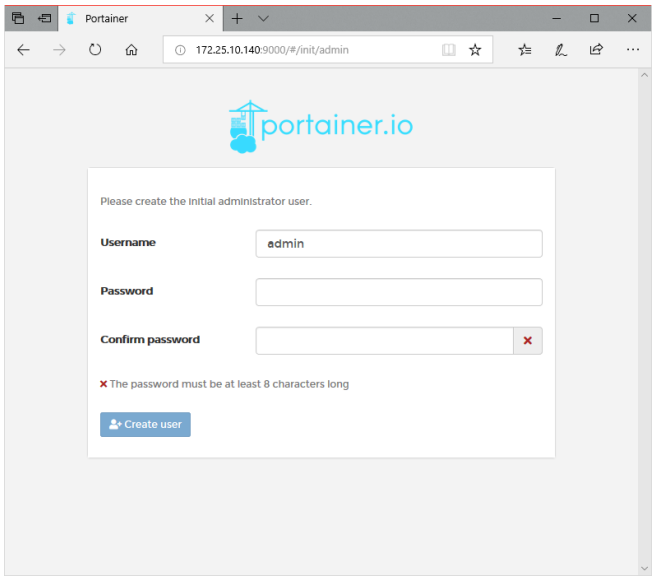
```
[root@c101086 ~]# docker images | grep portainer
portainer/portainer latest 47dbf4321bb4 4 weeks ago 10.7MB

# to create a new Portainer Container
[root@localhost ~]# docker run -d -v "/var/run/docker.sock:/var/run/docker.sock" -p
9000:9000 portainer/portainer

# you can access the web GUI via web browser
use <hostsystemIP>:9000

# you will be prompted to enter an admin password, choose manage docker where
portainer is running and connected
```

Initial portainer container instance password configured on the system is : password@1

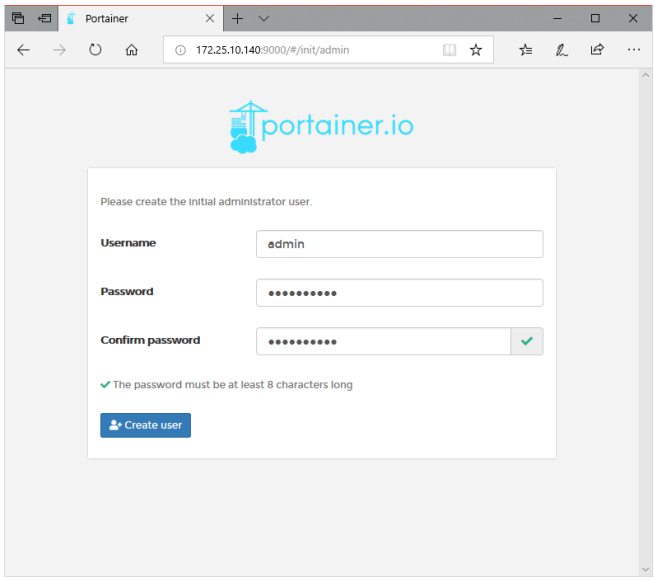
If the portainer container was removed, then the end user will have to supply a new password for the new container instance.

Initial Startup / Configure for a new instance of Portainer:

Type in a password for admin

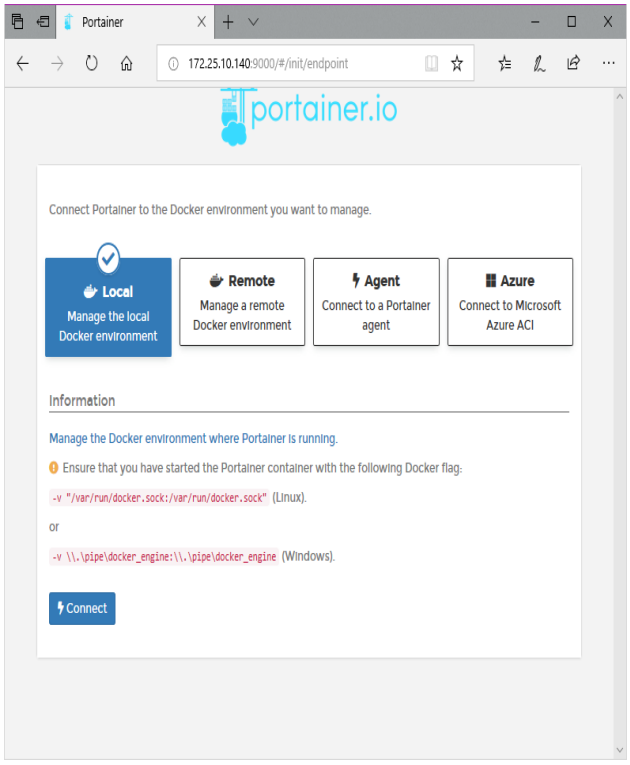


Click on Create User to continue

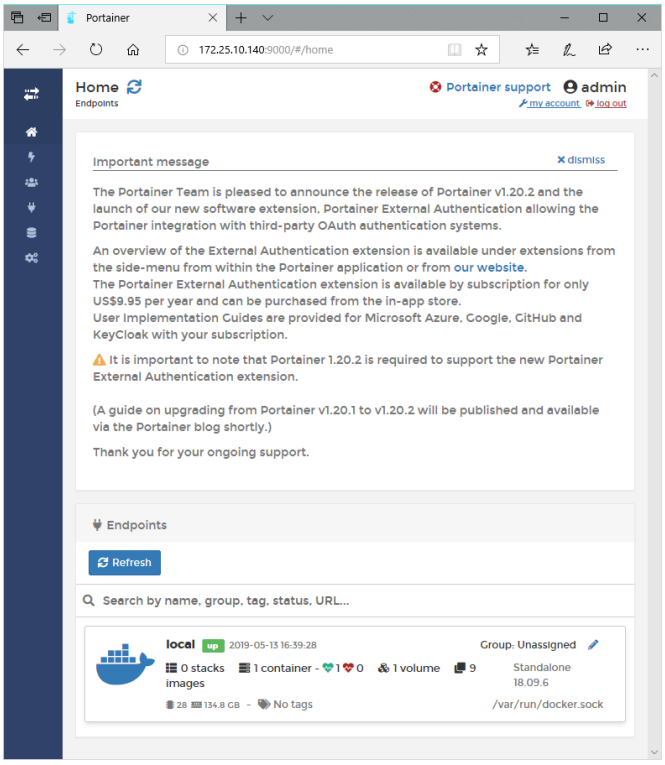


Portainer allow different way to connect to docker engine, Select Local to manage local docker setup.

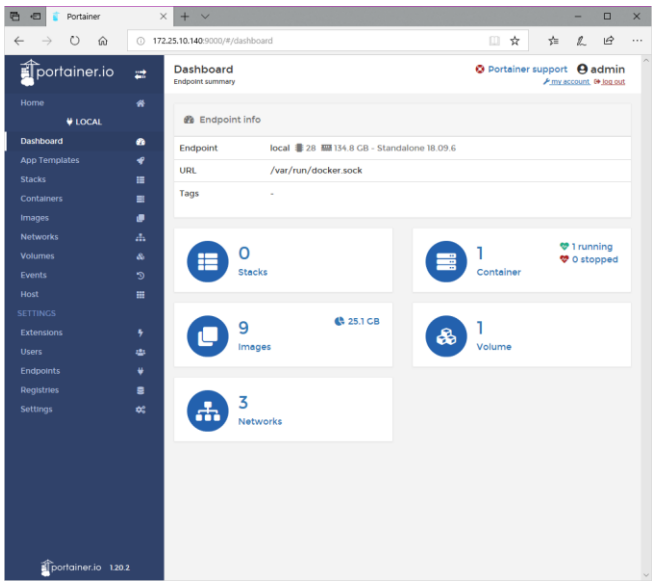
Click on Local and then click on Connect at the bottom.



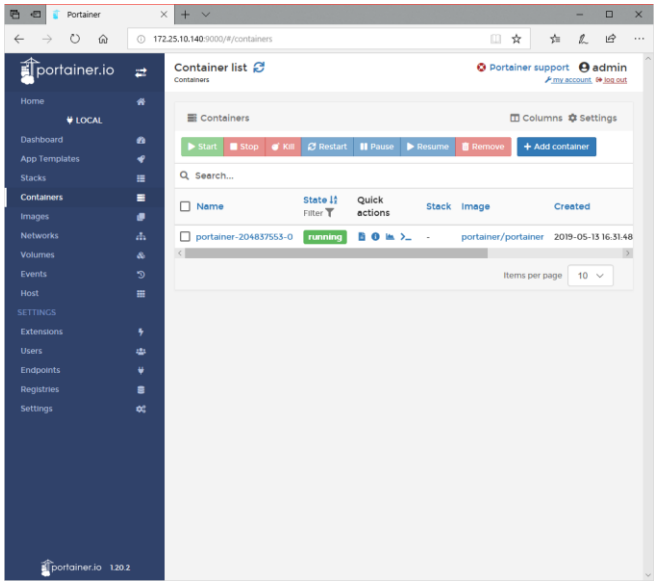
Portainer will confirm the selection. Click on local at the bottom of the screen to go to the dashboard.



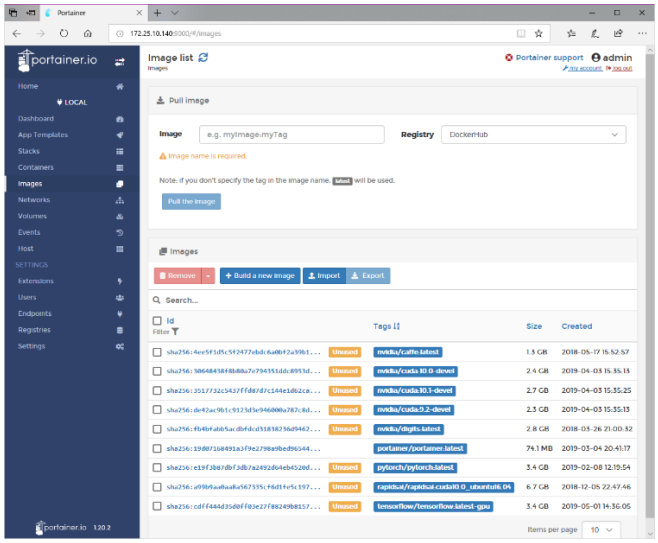
Dashboard View - Provides an overview of the container(s) running on the systems, along with the related volumes and network info.



Portainer Containers view – Overview of loaded container status, and control / manage of the containers



Portainer Images View - Overview of pulled images on the system, or download (pull) additional images available at the DockerHub Registry



Rapids Container and Notebook Server

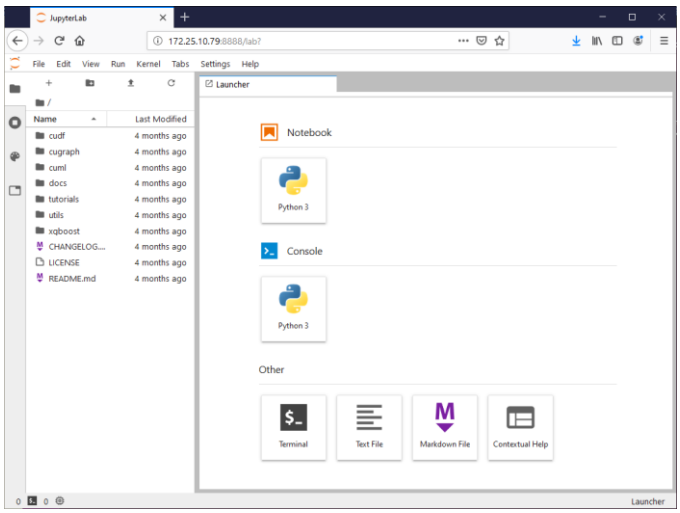
NOTE: This will run JupyterLab on port 8888 on your host machine.

Command:

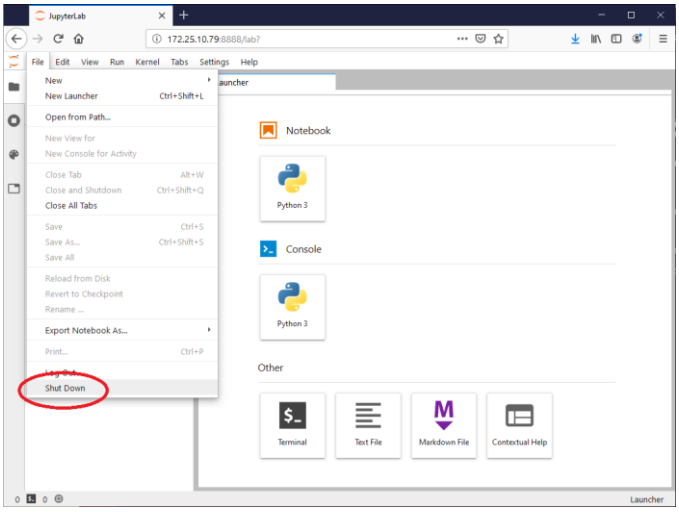
- docker run --runtime=nvidia --rm -it -p 8888:8888 -p 8787:8787 -p 8786:8786 nvcr.io/nvidia/rapidsai/rapidsai:cuda10.1-runtime-ubuntu18.04
- utils/start-jupyter.sh

- [root@c105017 ~]# docker run --gpus all --rm -it -p 8888:8888 -p 8787:8787 -p 8786:8786 nvcr.io/nvidia/rapidsai/rapidsai:cuda10.1-runtime-ubuntu18.04
-
- ## Starting jupyter service
- (rapids) root@712e75ae4a0e:/rapids/notebooks# bash utils/start-jupyter.sh
-
- jupyter-lab --allow-root --ip=0.0.0.0 --no-browser --NotebookApp.token=''
-
- [I 19:26:58.713 LabApp] Writing notebook server cookie secret to /root/.local/share/jupyter/runtime/notebook_cookie_secret
- [W 19:26:58.951 LabApp] All authentication is disabled. Anyone who can connect to this server will be able to run code.
- [I 19:26:58.964 LabApp] JupyterLab extension loaded from /conda/envs/rapids/lib/python3.6/site-packages/jupyterlab
- [I 19:26:58.964 LabApp] JupyterLab application directory is /conda/envs/rapids/share/jupyter/lab
- [W 19:26:58.966 LabApp] JupyterLab server extension not enabled, manually loading...
- [I 19:26:58.968 LabApp] JupyterLab extension loaded from /conda/envs/rapids/lib/python3.6/site-packages/jupyterlab
- [I 19:26:58.968 LabApp] JupyterLab application directory is /conda/envs/rapids/share/jupyter/lab
- [I 19:26:58.969 LabApp] Serving notebooks from local directory: /rapids/notebooks
- [I 19:26:58.969 LabApp] The Jupyter Notebook is running at:
- [I 19:26:58.969 LabApp] http://(712e75ae4a0e or 127.0.0.1):8888/
- [I 19:26:58.969 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
- [I 19:27:29.919 LabApp] 302 GET / (172.25.10.173) 1.71ms
- [W 19:27:30.730 LabApp] Could not determine jupyterlab build status without nodejs
- [W 19:27:30.925 LabApp] 404 GET /lab/api/workspaces/lab?1549654049120 (172.25.10.173): Workspace 'lab' ('lab-a511') not found
- [W 19:27:30.925 LabApp] Workspace 'lab' ('lab-a511') not found
- [W 19:27:30.926 LabApp] 404 GET /lab/api/workspaces/lab?1549654049120 (172.25.10.173) 1.45ms referer=http://172.25.10.206:8888/lab?

Screen shot - <host IP>:8888



To exit, select Shutdown from the File Menu:



Tensorflow:

NOTE: This will start Tensorflow container and switch to interactive console:

Command:

```
docker run --runtime=nvidia -it nvcr.io/nvidia/tensorflow:19.12-tf2-py3 bash
```

Please read the README.MD inside of the container for detail, or visit www.tensorflow.org for more information

Note: Docker version earlier then 19.03 with nvidia-docker2 installed will need to use

- -runtime=nvidia

flag for the NVIDIA GPU support in the container.

Docker version 19.03 and later with nvidia-container-toolkit installed will need to use -

- -gpus all

flag for the NVIDIA GPU support in the container.

For Additional Technical Support, please contact us at: www.exxactcorp.com/support