

# Deep Learning on Ubuntu with docker

## Dashboard

This page will offer instruction how to install docker on Ubuntu 16.04 LTS and configure several docker images for deep learning, such as NVIDIA DIGITS, TensorFlow or CNTK.

## Prerequisite

To be able to use docker and NVIDIA DIGITS the following bullets have to be fulfilled.

- GNU/Linux x86\_64 with kernel version > 3.10
- Docker >= 1.9
- NVIDIA GPU with Architecture > Fermi (2.1)
- NVIDIA drivers >= 340.29 with binary nvidia-modprobe

## NVIDIA driver

This was build based on a GTX 660 with driver version 367.57!

## Installing docker

To install docker follow the instruction on the [docker documentation](#) page or type in the basic commands listed below:

```
sudo apt-get update
sudo apt-get install apt-transport-https ca-certificates
sudo apt-key adv --keyserver hkp://p80.pool.sks-keyervers.net:80 --recv-keys
58118E89F3A912897C070ADBF76221572C52609D
echo "deb https://apt.dockerproject.org/repo ubuntu-xenial main" | sudo tee
/etc/apt/sources.list.d/docker.list
sudo apt-get updat
apt-cache policy docker-engine
sudo apt-get install linux-image-extra-$(uname -r) linux-image-extra-virtual
sudo apt-get install docker-engine
sudo service docker start
sudo docker run hello-world

# if user does not already exist:
sudo groupadd docker

sudo usermod -aG docker <$USER>
sudo systemctl enable docker
```

```
# logout and login again
docker run hello-world
```

## Installing NVIDIA GPU Support

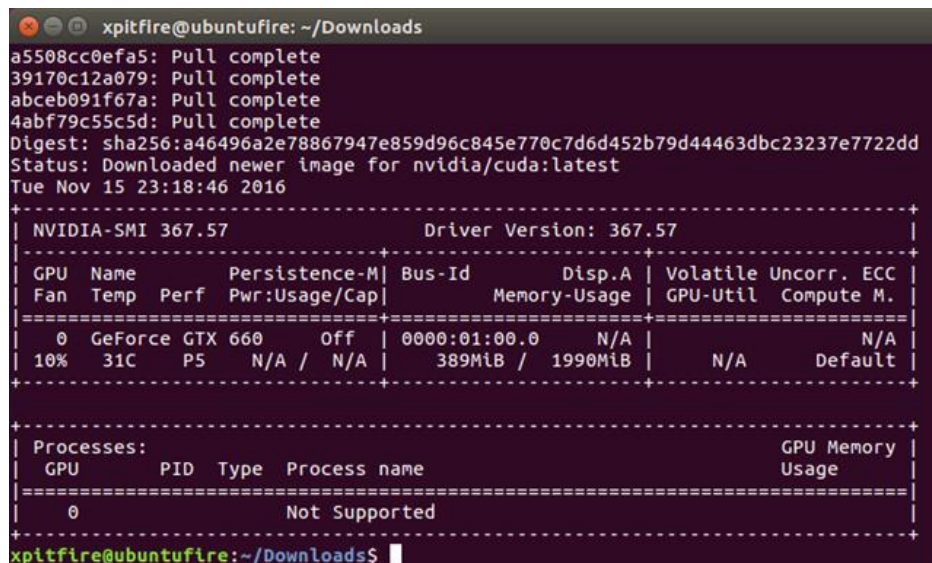
Type in the following commands to enable [docker CUDA GPU support](#):

```
# Install nvidia-docker and nvidia-docker-plugin
wget -P /tmp https://github.com/NVIDIA/nvidia-
docker/releases/download/v1.0.0-rc.3/nvidia-docker_1.0.0.rc.3-1_amd64.deb

sudo dpkg -i /tmp/nvidia-docker*.deb && rm /tmp/nvidia-docker*.deb

# Test nvidia-smi
nvidia-docker run --rm
nvidia/cuda nvidia-smi
```

The output should be similar to the following screenshot:



```
xpfitfire@ubuntufire: ~/Downloads
a5508cc0efa5: Pull complete
39170c12a079: Pull complete
abceb091f67a: Pull complete
4abf79c55c5d: Pull complete
Digest: sha256:a46496a2e78867947e859d96c845e770c7d6d452b79d44463dbc23237e7722dd
Status: Downloaded newer image for nvidia/cuda:latest
Tue Nov 15 23:18:46 2016

+-----+
| NVIDIA-SMI 367.57              Driver Version: 367.57          |
+-----+-----+
| GPU  Name          Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+
|  0  GeForce GTX 660      Off   | 0000:01:00.0  N/A   |         N/A           |
| 10%   31C   P5     N/A /  N/A | 389MiB / 1990MiB |    N/A   Default     |
+-----+-----+

+-----+-----+
| Processes:                      GPU Memory               |
| GPU       PID  Type  Process name                      Usage                    |
+-----+-----+
|  0                          Not Supported                      |
+-----+-----+
xpfitfire@ubuntufire:~/Downloads$
```

## Installing NVIDIA DIGITS docker container

The follow commands are based on [this](#) GitHub documentation:

```
mkdir ~/datasets
mkdir ~/digits-jobs

# Run DIGITS storing jobs in a host volume named digits-jobs and accessing
datasets from the datasets directory
nvidia-docker run -v ~/datasets:/usr/share/datasets --name nv-digits -d -p
5000:34448 -v ~/digits-jobs:/usr/share/digits/digits/jobs nvidia/digits
```

The above command will pull the NVIDIA docker image from its repositories and it afterwards it run the container. Now it should be possible to access NVIDIA DIGITS on the local machine using the browser.

**The service will listen on: 127.0.0.1:5000**

You should see something similar to the following screenshot:



## Installing Caffe support

To be able to use the NVIDIA DIGITS models outside of the docker container or to execute the scripts included in the GitHub repository it is necessary to setup the python and Caffe environment:

```
# install python and essential build libraries
sudo apt-get install python
sudo apt-get install --no-install-recommends build-essential cmake git
gfortran libatlas-base-dev libboost-all-dev libgflags-dev libgoogle-glog-dev
libhdf5-serial-dev libleveldb-dev liblmdb-dev libopencv-dev libprotobuf-dev
libsnpappy-dev protobuf-compiler python-all-dev python-dev python-h5py python-
matplotlib python-numpy python-opencv python-pil python-pip python-protobuf
python-scipy python-skimage python-sklearn python-matplotlib ipython ipython-
notebook python-pandas python-sympy python-nose
```

```
pip install image
pip install --upgrade pip
```

```
# the example location - can be customized
export CAFFE_ROOT=~/.caffe
git clone https://github.com/NVIDIA/caffe.git $CAFFE_ROOT
```

```
sudo pip install -r $CAFFE_ROOT/python/requirements.txt
```

```
# verify the installation of the requirements
cat $CAFFE_ROOT/python/requirements.txt | xargs -n1 sudo pip install
```

```
cd $CAFFE_ROOT
```

```
# build and install caffe
mkdir build
```

```
# if you want to enable the GPU support use cmake as show below, else jump
directly to STEP-2
```

```
cd build
# builds GPU supported Caffe binaries
cmake ..
make --jobs=4
```

```
# STEP-2: start build process
```

```
make all
make install
make runtest
```

```
export PYTHONPATH=$PYTHONPATH:~/caffe/python
```

## Download MNIST dataset using NVIDIA scripts or execute NVIDIA DIGITS pre-trained models

Navigate to the root directory of the cloned [NVIDIA DIGITS repository](#) or clone the repository as shown below:

```
git clone https://github.com/NVIDIA/DIGITS.git
```

The following command will download the MNIST dataset to a dataset subdirectory from the user home directory:

```
python -m digits.download_data mnist ~/mnist
```

If a model has been exported and packaged by the NVIDIA DIGITS Training, it can be used based on a python based Caffe script available within the NVIDIA DIGITS repository:

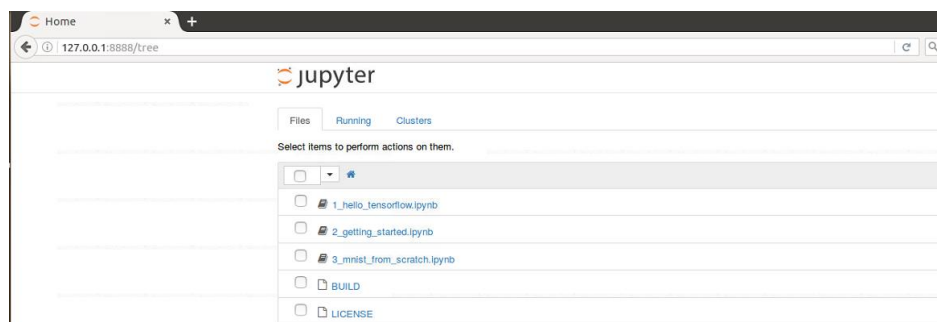
```
# <name-of-model> ... represents the exported NVIDIA DIGITS model
# /path/to/image/<test-image> ... represents the image you try to classify
# /path/to/repo ... represents the path to the cloned NVIDIA DIGITS
repository
cd /path/to/repo/DIGITS/examples/classification/
./use_archive.py --nogpu <name-of-model>.tar.gz ~/path/to/image/<test-
image>.png
```

## Installing TensorFlow docker container

To install the TensorFlow docker container with GPU support, type in the following command:

```
nvidia-docker run -it -p 8888:8888 gcr.io/tensorflow/tensorflow:latest-gpu
```

**The service will be listening on: 127.0.0.1:8888**



## Installing Microsoft CNTK docker container

To be able to use the Microsoft docker container it is necessary to first clone the [Microsoft CNTK GitHub](#) repository and build a new image:

```
# clone CNTK GitHub repository
git clone https://github.com/Microsoft/CNTK

# navigate to the docker directory, whereas
# /path/to/repo ... represents the path to the cloned CNTK repository
cd /path/to/repo/CNNTK/Tools/docker

# build the docker image
docker build -t cntk CNTK-GPU-Image

# install and run the new docker image
docker build -t ms/cntk:latest-gpu CNTK-GPU-Image
nvidia-docker run --name cntk -v ~/datasets:/usr/share/datasets -ti ms/cntk
bash
```