

Using Docker Images with Singularity

HOW THIS TUTORIAL WORKS

This tutorial is written in order to give a small insight in using Docker image files with Singularity. It will cover, connecting to cluster, pulling the image and starting a database for a small test script.

It is written as shell script itself and might just be executed on the cluster!

You may also just copy paste the commands for a better understanding.

Please check [this tutorial](#) if you are interested in creating an own image file.

Thx for reading
Jan Eberhardt

```
#!/bin/bash
# 0. Login on Frontend (you probably already did that)
# Use your TUB account and host gateway.hpc.tu-berlin.de
# ssh "<TUB account name>@gateway.hpc.tu-berlin.de"

# 1. Get Docker Image
# Go to your home directory and download the image via singularity.
# You must load the singularity module beforehand.
module load singularity/3.1.0

# Pulling docker images is done by Singularity's pull command. Source will
# be something like "docker://[package name]".
# Singularity will automatically download the latest version of the image
# and rewrite it to a Singularity image file (sif) as "[package
# name]_latest.sif".
# Therefore you will need write permission in your current working directory
# (which is why we changed into home).
cd
singularity pull "docker://mongo"

# 2. Create Python environment [if using Python]
# Most Python projects will use open source libraries installed by pip.
# Since normal users are not allowed to do so, it is recommended to install
# pip packages in user space or in a virtual python environment. We would
# discourage you from using user space for installation since most packages
# you will only use once in your life and it is therefore cleaner to get an
# unique environment for each project of yours.

# a) Load python module.
module load python/3.7.1

# b) Create the environment
py="$~/mongodb_venv"
python3 -m venv ${py}
```


