# Docker intro

## Running Docker

To do this workshop it easy to execute the commands as user root. In a production environment this is not adviced. Logon to your Ubuntu machine and become user root.

sudo su -

Before we start with anything, let's run the following command in the command line interface (CLI):

docker run hello-world

This runs a pre-existing image called hello-world on your Docker host, which runs and outputs some cute text.

#### Pulling images

On the internet there is a public registry: https://hub.docker.com/explore/ Here can you search for images. You can also search from the commandline with the following command:

docker search mysql

This searches for mysql images. The output should look somthing like this:

NAME DESCRIPTION STARS OFFICIAL AUTOMATED  
mysql MySQL is a widely used, open-source relation… 9146 [OK]  
mariadb MariaDB is a community-developed fork of MyS… 3244 [OK]  
mysql/mysql-server Optimized MySQL Server Docker images. Create… 676 [OK]  
centos/mysql-57-centos7 MySQL 5.7 SQL database server 69  
mysql/mysql-cluster Experimental MySQL Cluster Docker images. Cr… 62  
centurylink/mysql Image containing mysql. Optimized to be link… 61 [OK]

There is an important difference between the official images and non-official images. Everybody can upload images towards the Dockerhub, so also the evil hacker. The official images are trusted by Docker.

Running a docker pull means that your Docker Host downloads an image from the registry (but does not run it). So try running the following command:

docker pull ubuntu

This will pull the ubuntu image from the default Docker registry: https://hub.docker.com/explore/. You can also try pulling other images from the registry if you want!

Now that you have the ubuntu image, you can try running it:

docker run ubuntu

Notice that nothing happened. *This is on purpose* - we didn't specify a command, so all Docker did was to load up ubuntu, ran an empty command, then exited. So what does it look like when we want to run a command? Try:

docker run ubuntu date

This will load up the ubuntu image, then run the date command which outputs the date.

To see all the running and stopped containers on your system you can execute the following command:

docker ps -a

The -a option shows also the stopped containers. Take a good look at the output. The first collom is the container ID, within Docker this container ID is used a lot to address a specific container.

#### Seeing your Docker Images

The images are stored in the local registry. You can check the Docker images you have by running images:

docker images

This should list something similar to:

REPOSITORY TAG IMAGE ID CREATED SIZE  
ubuntu latest ccc6e87d482b 5 weeks ago 64.2MB  
hello-world latest fce289e99eb9 13 months ago 1.84kB

#### Let's run something in Docker

Okay. Now let's try running a website - say, an Nginx application that just serves the default page. For this, all we have to do is run:

docker run nginx:latest

Notice that we didn't have to do a docker pull. What essentially happens is shown in the output of the command:

# docker run nginx  
Unable to find image 'nginx:latest' locally  
latest: Pulling from library/nginx  
bc51dd8edc1b: Pull complete  
66ba67045f57: Pull complete  
bf317aa10aa5: Pull complete  
Digest: sha256:ad5552c786f128e389a0263104ae39f3d3c7895579d45ae716f528185b36bc6f  
Status: Downloaded newer image for nginx:latest

As per the above, Docker tries to see if the image you're trying to run exists locally - then pulls from somewhere else if it does not exist. The *somewhere else* is the Docker Hub we saw earlier - it just grabs the matching version off that.

But wait, it seems to be stuck? Not really. Docker is just running on attached mode, so we also need to run Docker in a *detached mode* it looks like. Just type *Ctrl-C* to cancel the running container, and rerun it with as the following:

docker run -d -P --name windesheim nginx

-d runs Docker in *detached mode*, and -P publishes all exposed ports so we can access it. windesheim is what we're naming the container - feel free to change it to any name you want. We can then find out how to access the site by running:

docker port windesheim

Which will give us something like:

80/tcp -> 0.0.0.0:32768

This basically means that the service is accessible from port 32768 of your machine. You should be able to curl to this website from the commandline:

curl http://localhost:32768

You can see all running containers with the docker ps command. The -a option shows also all stopped containers. Look at the DockerID. This ID is used a lot with docker to address a container.

docker ps -s

Finally, stop the container by running:

docker stop windesheim

#### Docker Run Shell Within Ubuntu image

Note that the Docker image you're running also contains other things. It has a small slice of the operating system, and this slice is actually available to you. You can access the command line inside the nginx container we were running earlier by running:

docker run -it ubuntu /bin/bash

The -it flag attaches you to an interactive "terminal" within the container. /bin/bash is a shell that exists within the container. You are now in the container. Look at the processes that are running in the container with the following command:

ps -ef

In the output you can see that there are only 2 processes are running (the bash command and the ps command). The processes of other containers and the processes of the main system are not visable.

There you go! You can pull down and run containers and interact with them.

#### Docker filesystem & volumes

When a container is started, it alway uses the clean image. Any changes you made or files you add on the container are gone when you start a new container.

To see the filesystem changes that are made related to the original image you can use the docker diff command.

docker diff [DOCKER ID]

To persist data you need to mount a volume to the docker container, with the following command the /tmp directory is mounted to the container on path /data. You have an interactive shell (the -it option) to this ubuntu image

docker run -it -v /tmp:/data ubuntu /bin/bash

You can see the /tmp directory is now mounted to /data. If you add files to /data in the container, they are placed on /tmp on the host.

#### Exercise

Now, what you're going to find is that there are lots of Docker containers out in the wild. Now that you know how to pull down and run images, I've got a challenge for you: why not see if you can find and run the wordpress image from https://hub.docker.com/\_/wordpress ?