



UNIL | Université de Lausanne

Week 9(1) - Big-Scale Analytics

Introduction to docker

Docker is a set of Platform as a Service (PaaS) products that uses OS-level virtualisation to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels. All containers are run by a single operating system kernel and therefore use fewer resources than virtual machines (definition from Wikipedia). The software that hosts the containers is called **Docker Engine**.

Docker can package an application and its dependencies in a virtual container that can run on any Linux server. This will provide **flexibility** and **portability** enabling the application to be run in various locations, whether on-premises (e.g, your computer), in a public cloud, or in a private cloud. Therefore, you don't need to worry about dependencies if you want to run your application in different locations. Moreover Docker containers are lightweight and they use fewer resources than virtual machine, so you can run several containers on a single server simultaneously.

Installing Docker Desktop

Please refer to the [Docker documentation](#) to see how to install Docker Desktop. Docker Desktop is an easy-to-install application for your Mac or Windows environment that enables you to build and share containerized applications and microservices. Docker Desktop includes Docker Engine, Docker CLI client, Docker Compose, Notary, Kubernetes, and Credential Helper.

Running a Docker image

In this section we will see how to run a docker image. Docker images are templates which are used to create Docker containers. They contain all the files, dependencies, etc required to run a container. Container is a running instance of an image. You can find many different pre-built Docker images in [Docker Hub](#). In today's lab we are going to run a Docker image which will run a Python script.

Open the terminal in your computer (cmd in windows) and run the following command. You will be able to see all of the docker images you have on your system:

```
docker images
```

Next, download the Docker image from the following link:

<https://drive.switch.ch/index.php/s/Jc2vjxsBDMM0GHI>

Load the downloaded image, which is a “.tar” file, using the following command:

```
docker load < “image_name”
```

Now if you run “docker images” again, you will see the new image in the list. Next you will start a container based on this image by running the following:

```
docker run “image_name”
```

You will see that as the container starts, the python script will be executed and you will see the result in your terminal page. The script will randomly select a Wikipedia article and returns the title and the summary of the article. As you can see, the container stops as soon as the execution of the Python script is finished. Can you explain why this happens?

You can also see the list of docker containers running on your computer:

```
docker ps
```

And if you want to see the list of all containers, whether running or stopped, run:

```
docker ps -a
```

For deleting a container simply run:

```
docker rm “container_id”
```

And for deleting images, run:

```
docker rmi “image_name”
```

Running an image in interactive mode

Follow this [12 minutes tutorial](#) video that explains how to pull an image from Docker Hub and run it in interactive mode.

Building a Docker image — for next week

Next week, we will discuss how to create and build Dockerfile which is a text file with instructions to build an image. Check out [this tutorial video](#) on creating and building Dockerfile in order to get prepared for next week. Moreover, if you are interested in dockerizing Python applications, check out [this tutorial](#). We will have more on this topic next week.