Lab Containerization

Server OS

Bachelor Toegepaste Informatica
Graduaat Netwerken

Auteur : Geert Coulommier

Design, Technologie en Gezondheidszorg

Table of contents

Table of contents 2
1. Prerequisites 3
2. How to use this lab 3
3. Docker 4
3.1 Local installation of Docker in CentOS 4
3.2 Getting started with Docker 5
3.2.1 Tutorial 5
3.2.2 Exercise 5
3.3 Building and working with images 5
3.3.1 Tutorial 5
3.3.2 Exercise 6
3.4 Working with volumes 6
3.4.1 Tutorial 6
3.4.2 Exercise 6
3.5 Docker Compose 7
3.5.1 Tutorial 7
3.5.2 Exercise 7
Kubernetes 8
3.6 Tutorial 8

1. Prerequisites

What you need for this lab

Depending on your personal environment.

- Internet connection
- online Docker environment in a webbrowser:
    - Docker ID
    - https://training.play-with-docker.com/
      Good interactive tutorials with ample explanation.
    - https://labs.play-with-docker.com/
      Interactive Docker Playground where one can test Docker without the need of an actual Docker installation.
    - https://www.docker.com/101-tutorial/
      Run the ”`docker run -dp 80:80 docker/getting-started:pwd`” command in a Docker environment (like the Docker Playground) and follow the tutorials to learn about Docker.

- Windows client OS (e.g. Windows 11) Docker environment:
    - https://docs.docker.com/docker-for-windows/install/
    - Windows client OS in a Professional or enterprise version
    - Computer with virtualization enabled in the BIOS/UEFI
    - Windows features enabled:
        - Containers
        - Hyper-V
    - Docker Desktop
- Windows Server OS Docker environment:
    - Windows features enabled:
        - Containers
        - Hyper-V
    - Execute commands:

```
Install-Module -Name DockerMsftProvider -Repository PSGallery –Force
Install-Package -Name docker -ProviderName DockerMsftProvider
```

- MAC OS Docker environment:
    - https://docs.docker.com/desktop/install/mac-install/
- Linux Docker environment:
    - Dependent on Distribution
    - e.g. Ubuntu:
        - https://docs.docker.com/engine/install/ubuntu/

2. How to use this lab

This lab will use a combination of labs from play with Docker and additional exercises.

The tutorial will give a step-by-step tutorial to get you acquainted with all the concepts and commands. To get the most out of these tutorials, make sure to read the text carefully so that you understand what is said, and follow the commands and instructions to the letter so that you get the same final result. Although it is possible to just click most of the commands to make them execute in the cli-environment, it is advised to type the commands by hand to better memorize them.

The exercise will test your newly learned skills by presenting you with a desired end result, which you can achieve by using all of the material from the tutorial and previous chapters. This is more a personal playground, so feel free to experiment with different commands and approaches to attain the requested result. Also try to repeat the commands and exercises a few times to get a good feel for them. For Docker exercises, this second step can be done in a Docker environment of choice such as Docker Desktop or Docker installation in a Linux VM. If you don't have one available, you can always use https://labs.play-with-docker.com/, which comes highly recommended as a freely available testing and learning environment, with its only caveat that everything will be erased after 4 hours.

3. Docker

3.1 Local installation of Docker in CentOS

Play with Docker and its tutorials are of course very practical for testing and learning how to work with Docker, but for a more realistic simulation, we can use a local installation of Docker on our own server. To accomplish this, we will use a VM. See the "Server OS-OS II - labo - locations, logincredentials and port forwardings.pdf" document and added documents to connect to this VM.

To install Docker on this VM, use the tutorial on https://docs.docker.com/engine/install/ and select the Server (not Docker Desktop) procedure that matches your VM OS/distribution.

Additional notes for installation on CentOS:

- Always try to connect to a VM remotely using RDP or SSH instead of using the console view in Virtual Machine Manager. As we are working with a Linux VM, we will need SSH. As a client you can use putty, the ssh-client built into Windows (try to open Powershell cli), or best of all, use the Windows terminal with all its benefits, among which the ability to add connections directly to the configuration file.
- There might be a discrepancy in the dependencies between the provided docker-ce package and the dockerd.io package. You can use the `--allowerasing` option to install an older version of Docker with all dependencies corresponding with the below command.
  ```
  sudo yum install docker-ce docker-ce-cli containerd.io --allowerasing
  ```
- The procedure as mentioned on Docker docs for installing Docker on CentOS does mention the need for starting the Docker service through the command `sudo systemctl start docker`, but the service will also need to be enabled so that it will automatically start when the VM starts. You can use the command `sudo systemctl enable docker` to achieve this.

Once ready, you can for instance run the httpd container mapping internal port 80 to port 80 on the host and testing it in a web browser with the IP-address of your VM network mentioned in "Server OS-OS II - labo - locations, logincredentials and port forwardings.pdf".

It is also a good idea to check out https://docs.docker.com/install/linux/linux-postinstall/ once the installation is successful. Among other things, it describes the procedure to allow a regular useraccount to run Docker commands without the need of `sudo` everytime.

3.2 Getting started with Docker

3.2.1 Tutorial

This lab familiarizes you with the basic Docker concepts and commands.

https://training.play-with-docker.com/ops-s1-hello/

3.2.2 Exercise

- Display all images
- Pull the Ubuntu image
- Display all images
- Display all running containers
- Display all containers
- Run the hello-world image
- Display all images
- Display all containers
- Run an ubuntu container interactively with a /bin/bash cli
- Exit
- Display all containers
- Run an ubuntu container interactively with a /bin/bash cli
- Perform some Linux commands (like top, ls -hal /etc) in the container to check that you really have full cli control of the container
- Press the keys CTRL + P + Q simultaneously (exit shell without stopping the container)
- Display all containers
- Enter the container interactively with a /bin/bash cli
- Exit
- Display all containers
- Run an ubuntu container in the background (detached)
- Display all containers
- Stop this container
- Display all containers
- Start this container
- Display all containers
- Remove all containers and images

3.3 Building and working with images

3.3.1 Tutorial

This lab will show you how to work with images and how to build one using a Dockerfile. Warning: when copying from the assignment into vi in the Docker Playground, it is possible that the first characters of the text are not copied over correctly. Recheck the pasted text.

https://training.play-with-docker.com/ops-s1-images/

3.3.2 Exercise

- Verify all steps below with the appropriate commands. See the commands in exercise 3.2.2 for inspiration.
- Clone the bulletinboard app from GitHub (if GitHub is not yet available in your environment, you'll need to install it first):
  `git clone -b v1 https://github.com/docker-training/node-bulletin-board`
- Inspect the Dockerfile and find all the different elements represented in them.
- Build an image using the Dockerfile with the name 'bulletinboard:1.0"
- Run a new container based on this image with these options:
  - Detached
  - Internal port (find it in the online documentation of the app) mapped to port 80 on the host
  - Name the container "bb"
- Test the app by opening it in a web browser
  - When working locally, use "localhost" to identify the computer
  - When working on a remote computer, use the remote computer's IP or DNS-name
  - When working with play with Docker, use the link provided on the top of the screen
- Remove all containers and images

3.4 Working with volumes

3.4.1 Tutorial

This lab will show you how to persist data by using volumes. Note: in this tutorial, the option `-v` is used, which is the older way of working with volumes. It still works, but it is advised to use the `--mount` option instead for greater versatility.

https://training.play-with-docker.com/docker-volumes/

3.4.2 Exercise

- Pull the CentOS image.
- Run a container based on the CentOS image interactively.
- Create a text file with content in "Hello world!" in /root in the container.
- Exit.
- Stop the container.
- Start the container with an interactive shell and notice that the file is still there in the container.
- Exit.
- Remove the container.
- Run a new container based on the same CentOS image interactively.
- Check for the existence of the file in the container: it should not be there anymore.
- Create a volume named "myvol".
- Run a container based on the CentOS image with parameters:
  - Interactive

    - The volume "myvol" mapped to internal folder /root
- In the container, create a text file with content "Hello world!" in /root.
- Exit.
- Remove the container.
- Run a new container based on the CentOS image with parameters:
    - Interactive
    - The volume "myvol" mapped to internal folder /root
- Check that the file is still there in the container. It should be, even after the container was removed.
- Exit
- Remove the container.
- Run a container based on the CentOS image with parameters:
    - Interactive
    - A local folder from the host mapped to internal folder /root
- In the container, create a text file with content "Hello world!" in /root.
- Exit.
- Remove the container.
- Run a new container based on the CentOS image with parameters:
    - Interactive
    - A local folder from the host mapped to internal folder /root
- Check that the file is still there in the container. It should be, even after the container was removed.
- Remove all volumes, containers and images.

3.5 Docker Compose

This lab will show you how to use Docker Compose, a tool to combine the configuration and start-up parameters of multiple containers in one docker-compose.yml file.

3.5.1 Tutorial

In this tutorial, you will learn the basics of using Docker Compose to build and run a Docker stack of applications. If you are not yet familiar with the fundamentals of Docker Compose, it is advised to first read the information on this webpage about this useful tool.

Go to https://www.docker.com/101-tutorial/ and follow the "Play with Docker" procedure. Once you have started the new instance in the Docker Playground, started the new container with the "`docker run -dp 80:80 docker/getting-started:pwd`" command and clicked on the "80" button on top of the screen, a new tab in your webbrowser application should open. This is the actual tutorial description (running in a container in your online Docker instance). Here, select "Using Docker Compose". This tutorial refers to previous exercises. The only thing that needs to be done to make this work, is download and unzip the app.zip file from the "Our Application link on the left of the tutorialpage. Make sure to create the docker-compose.yml file in the app folder and run the `docker compose up -d` command from inside the same folder.

3.5.2 Exercise

Although this exercise is based on an online guide, it requires more from the user to make it work than above tutorial and is therefore an interesting exercise to practice working with

Docker Compose. To get the most out of the exercise, make sure to read all instructions and information comprehensibly.

Note: as stated in the in the online exercise, docker-compose needs to be installed in your environment. If this is not yet the case, follow the instructions as stated here.

Note: The exercise assumes you are executing the steps on a Docker Desktop host. If this is not the case, you will have to adapt to your environment. E.g., on the EhB-VM, you will have to use the IP-address assigned to you and change to port to one that is forwarded to the CentOS-VM. So change 5000:5000 to 443:5000 and use the uri http://10.3.50.[x]:443 for testing with [x] being the IP octet specific for you.

Actual exercise: https://docs.docker.com/compose/gettingstarted/

Kubernetes

3.6 Tutorial

The online lab which can be found in the link below, will familiarize you with the basics of Kubernetes, the main orchestration tool for (Docker) orchestration. It uses minikube, which is a small install of the Kubernetes platform, created specifically for Kubernetes learning environments. If you like to play around with Kubernetes in your own environment, you can always do a local install of minikube, or use the Kubernetes playground, which can be found here: https://labs.play-with-k8s.com/, but for this exercise follow the Kubernetes basics online tutorial from the link at the bottom of this chapter.

Some tips for using this online tutorial:

- Make sure to read the concepts well before starting the interactive tutorial to make sure you know what you are doing.
- Some commands can take several minutes to execute, especially starting up Kubernetes at the start of each module. Be patient and, if possible, continue reading about the next command(s).
- Not all browsers work well with the Kubernetes playground. If one doesn’t seem to work to well, try another.
- Finish all modules.
- This is an interactive environment, which means you can do more than just the commands provided in the tutorial. Sometimes it might be useful to run commands several times to monitor the evolution of a process, especially those that get information.

https://kubernetes.io/docs/tutorials/kubernetes-basics/